

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:
Ingenieros de Sistemas**

TEMA:

**DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DOMÓTICO PARA
CONTROLAR DISPOSITIVOS CONECTADOS MEDIANTE UNA RED DE
SENSORES, A TRAVÉS DE UN SERVIDOR DOMÉSTICO.**

**AUTORES:
DIEGO FERNANDO GUALLPA ALPALA
JOSÉ GREGORIO LAGOS POZO**

**TUTOR:
MANUEL RAFAEL JAYA DUCHE**

Quito, agosto del 2019

CESIÓN DE DERECHOS DE AUTOR

Nosotros, GUALLPA ALPALA DIEGO FERNANDO, con documento de identificación N° 1720119849, y LAGOS POZO JOSÉ GREGORIO, con documento de identificación N° 1724126642, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de grado titulado “DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DOMÓTICO PARA CONTROLAR DISPOSITIVOS CONECTADOS MEDIANTE UNA RED DE SENSORES, A TRAVÉS DE UN SERVIDOR DOMÉSTICO.” mismo que ha sido desarrollado para optar por el título de INGENIEROS DE SISTEMAS, en la Universidad Politécnica Salesiana, quedando la universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada.

En concordancia, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



.....
DIEGO FERNANDO
GUALLPA ALPALA

CI: 1720119849



.....
JOSÉ GREGORIO
LAGOS POZO

CI: 1724126642

Quito, agosto del 2019

DECLARATORIA DE COAUTORÍA DEL TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico realizado por Diego Fernando Gualpa Alpala y José Gregorio Lagos Pozo, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, agosto 2019



.....
MANUEL RAFAEL JAYA DUCHE
CI: 1710631035

DEDICATORIA

Primeramente, agradezco a Dios por haberme guiado durante toda esta etapa de mi vida y lo más importante haberme dado toda la sabiduría e inteligencia para culminar con esta importante meta en vida profesional, A mis padres que siempre han sido mi motor fundamental por darme su, amor, comprensión, apoyo incondicional en cada momento difícil de mi vida, y por ayudarme con todos los recursos para poder estudiar. Han sembrado en mí valores, mis principios, mi perseverancia y mi coraje para llegar a todas las metas que me proponga.

Todo este esfuerzo y dedicación ha sido gracias a ellos.

Diego Gualpa

Agradezco primero a dios por guiarme en el camino hacia el bien en mis estudios y en todas las etapas de mi vida brindándome todos los valores necesarios y la sencillez completa para actuar ante diversas situaciones. A mis padres por aconsejarme y corregirme mis errores en mi etapa estudiantil a la vez agradecer por brindarme su amor, cariño y comprensión. A mi hermana que ha sido mi guía para completar las metas propuestas y de igual manera por todos sus consejos para poder llevar a cabo la finalización del proyecto.

José Lagos

AGRADECIMIENTOS

Este proyecto de tesis es el resultado del esfuerzo en conjunto por varios años. Por esto agradecemos en primer lugar a Dios por permitirnos culminar con éxito nuestra carrera profesional, a nuestros padres por depositar toda su confianza en nosotros, a nuestro tutor Ing. Manuel Rafael Jaya Duche, por su paciencia y conocimientos aportados durante la realización del proyecto de tesis y a todos los docentes de la universidad que, de alguna forma, son parte de su culminación. Nuestros sinceros agradecimientos.

Diego Fernando Gualpa

José Gregorio Lagos Pozo

ÍNDICE

INTRODUCCIÓN	1
JUSTIFICACIÓN	1
OBJETIVOS	2
OBJETIVO GENERAL	2
OBJETIVOS ESPECÍFICOS	2
METODOLOGÍA	3
MÉTODO INDUCTIVO	3
MÉTODO DEDUCTIVO	3
CAPITULO 1	4
1.1. DOMÓTICA	4
1.2. BENEFICIOS DE LA DOMÓTICA	4
1.2.1. CONFORT	4
1.2.2. SEGURIDAD	5
1.2.3. ECONOMÍA	5
1.3. COMPONENTES DE UN SISTEMA DOMÓTICO	5
1.3.1. CENTRAL DE GESTIÓN	5
1.3.2. SENSORES	5
1.3.3. CONTROLADORES	10
1.3.4. ACTUADORES	11
1.4. RED DE SENSORES INALÁMBRICOS	14
1.5. TIPOS DE ARQUITECTURA	15
1.5.1. MEDIOS DE INTERCONEXIÓN	17
1.6. SISTEMAS EMBEBIDOS	18
1.6.1. RASPBERRY PI 3	18
1.6.2. PUERTOS GPIO	22

1.6.3. SISTEMA OPERATIVO RASPBIAN	24
1.7. ARQUITECTURA DE UN SISTEMA EMBEBIDO	25
1.7.1. NODEMCU	25
1.7.2. MÓDULO WIFI ESP8266	27
1.8. LENGUAJES DE PROGRAMACIÓN WEB	28
 CAPITULO 2	 33
2.1. HARDWARE PARA EL CONTROL DOMÓTICO	33
2.1.1 NODEMCU ESP8266	34
2.1.2 SENSORES Y ACTUADORES	34
2.1.3 GESTOR DE BASE DE DATOS MySQL	37
2.1.4 INTERFAZ GRÁFICA	37
2.2. SOFTWARE PARA EL CONTROL DOMÓTICO	38
2.3. PROTOCOLO DE COMUNICACIÓN	39
2.3.1. SERVIDOR WEB	39
 CAPITULO 3	 41
3.1. ESTUDIO DEL PROYECTO	41
3.2. ELECTRÓNICO	43
3.2.1. MODULO CON RELAY	44
3.3. SERVIDOR CENTRAL	56
3.4. SOFTWARE	58
3.4.1. DIAGRAMA DE COMUNICACIÓN DE HARDWARE	58
3.4.2. ANÁLISIS DE SOFTWARE	59
3.4.3. ARQUITECTURA DEL SISTEMA	59
3.4.4. DIAGRAMA DE CASOS DE USO	60
3.4.5. INTERFAZ DE USUARIO	63
3.4.6. INICIO DE SESIÓN	65
3.4.7. CRUD DE USUARIOS	66
3.4.8. PANTALLA DE ESTANCIAS	67
3.4.9. PANTALLA DE ELEMENTOS	68
3.4.10. PANTALLA DE CONTROL DE LUCES	69

3.4.11. PANTALLA DE SENSOR DE MOVIMIENTO	70
3.4.12. PANTALLA DE SENSOR DE HUMO	71
3.4.13. CONTROL DE PERSIANAS	72
3.4.14. PANTALLA DE AUDITORIA	72
 CAPITULO 4	 74
 PRUEBAS Y RESULTADOS	 74
 4.1. PRUEBAS DE HARDWARE	 74
4.2. PRUEBAS DE SOFTWARE	82
4.3. PRESUPUESTO	91
 CONCLUSIONES	 92
 RECOMENDACIONES	 93
 LISTA DE REFERENCIAS	 94

ÍNDICE DE FIGURAS

Figura 1. Tipos de Sensores	6
Figura 2. Estructura del sensor de movimiento	7
Figura 3. Sensor de Humo con salida digital y salida analógica	10
Figura 4. Características del sensor MQ135	10
Figura 5. Tarjetas Controladoras	11
Figura 6. Conexiones del Servo Motor	12
Figura 7. Características del actuador Tower ProSG90	12
Figura 8. Estructura del Relay	14
Figura 9. Estructura básica de una red de sensores inalámbrica.	15
Figura 10. Arquitectura Domótica Centralizada.....	16
Figura 11. Arquitectura Domótica Distribuida.....	16
Figura 12. Arquitectura Domótica Mixta.....	17
Figura 13. Características de la Raspberry Pi 3.....	19
Figura 14. Características Raspberry PI 3 B y B+.....	20
Figura 15. Raspberry Pi 3 Modelo B	21
Figura 16. Raspberry PI 3B+.....	21
Figura 17. Puertos GPIO en Raspberry Pi 3.....	23
Figura 18. Sistema Operativo Raspbian para Raspberry Pi.	24
Figura 19. Pinout de una placa NodeMCU.	26
Figura 20. Módulo ESP8266.....	27
Figura 21. Especificaciones del Micro controlador ESP8266.....	28
Figura 22. Funcionamiento de JavaScript	29
Figura 23. Interfaz PHP	30
Figura 24. Estructura del IDE Arduino 1.8.9	31
Figura 25. Arquitectura del Sistema Domótico	33
Figura 26. Características de los sensores y actuadores utilizados.....	35
Figura 27. Conexión de los elementos al NodeMCU ESP8266.....	35
Figura 28. Estructura del proyecto.....	42
Figura 29. Librerías de conexión por WiFi.	43
Figura 30. Conexión Wifi del NodeMCU.....	44
Figura 31. Imagen de Modulo Relay conectado al MCU	45
Figura 32. Inicialización de pines para el Modulo Relay	45

Figura 33. Programación para activar los Modulo Relay.	46
Figura 34. Código de la petición hacia el servidor web y su respuesta.	46
Figura 35. Valor de baudios para la comunicación de NodeMCU.....	47
Figura 36. Esquema de conexión del MCU con el sensor PIR y el módulo Relay	48
Figura 37. Asignación de valores de pines para la conexión del sensor PIR y el módulo Relay.....	48
Figura 38. Programación del sensor PIR.	49
Figura 39. Código de petición GET hacia el servidor web.	50
Figura 40. Conexión de NodeMCU con el servomotor	51
Figura 41. Código que asigna el pin de conexión para el servomotor.	51
Figura 42. Código que realiza la rotación del motor.	52
Figura 43. Código que envía la petición http al servidor.	53
Figura 44. Conexión del MQ-135 con NodeMCU	54
Figura 45. Lectura de pin analógico para coeficiente de gases.	54
Figura 46. Código de funcionamiento del detector de gases.....	55
Figura 47. Código de envío Http.	56
Figura 48. Comandos de instalación de PHP.	57
Figura 49. Sentencias de inicialización Apache2 y Mysql.....	57
Figura 50. Esquema de comunicación.	59
Figura 51. Arquitectura del sistema domótico.	60
Figura 52. Diagrama de caso de uso general.....	61
Figura 53. Caso de uso administrador.....	62
Figura 54. Caso de uso controlador	63
Figura 55. Interfaz de usuario inicial	64
Figura 56. Esquema de navegación del sistema	64
Figura 57. Pantalla de Login	65
Figura 58. Pantalla de inicio	66
Figura 59. CRUD de administración de usuarios	67
Figura 60. Interfaz estancias.....	68
Figura 61. Pantalla elementos.....	69
Figura 62. Pantalla control de luces.....	70
Figura 63. Pantalla de visualización de estado sensor de movimiento.	71
Figura 64. Pantalla de visualización de estado de detección de humo.	71
Figura 65. Interfaz de persianas.....	72

Figura 66. Visualización de auditoria	73
Figura 67. Toma de datos del sensor iluminación.	74
Figura 68. Sensor de luz diseñado.	75
Figura 69. Toma de datos del sensor de movimiento	75
Figura 70. Sensor de movimiento conectado al NodeMCU	76
Figura 71. Interfaz Web	77
Figura 72. Toma de datos del sensor de movimiento	78
Figura 73. Sensor de Humo conectado al NodeMCU	78
Figura 74. Interfaz Web	79
Figura 75. Toma de datos del servomotor.....	80
Figura 76. Actuador servomotor conectado al NodeMCU	80
Figura 77. Interfaz Web	81
Figura 78. Ingreso al sistema domótico	82
Figura 79. Validación de campos vacíos en usuario.....	83
Figura 80. Usuarios ingresados	84
Figura 81. Consulta de usuarios.	84
Figura 82. Listas de usuarios existentes.....	85
Figura 83. Modificación de usuarios	86
Figura 84. Verificar usuario modificado	87
Figura 85. Eliminación de usuarios.	88
Figura 86. Auditoría usuarios eliminados.	88
Figura 87. Ingreso de estancia con estado activo.	89
Figura 88. Ingreso de elemento.	90

ÍNDICE DE TABLAS

Tabla 1. Características del sensor PIR.....	9
Tabla 2. Características Relay	14
Tabla 3.Especificaciones del NodeMCU ESP8266.....	26
Tabla 4. Especificación de base de datos	37
Tabla 5. Especificaciones de la interfaz grafica	38
Tabla 6. Presupuesto.	91

Resumen

En el presente proyecto se desarrolló un sistema domótico de hardware y software, el cual permitió automatizar una vivienda, con el fin de facilitar los hábitos cotidianos del usuario, de tal manera que solo con manipular un pc o un teléfono celular pueda tener el control de su casa, claro está que esto dependerá de los servicios que se programaron e instalaron dentro del hogar.

El objetivo fue tener el control de una vivienda, mediante una conexión inalámbrica para manipular dispositivos electrónicos utilizando dispositivos móviles o una computadora.

Para la realización del proyecto se utilizaron dispositivos electrónicos como micro controladores, sensores, actuadores que enviaron información al servidor principal. El servidor la procesó y emitió órdenes a los distintos dispositivos electrónicos, mediante la utilización de peticiones HTTP GET y HTTP POST; cabe recalcar que el sistema desarrollado funcionó bajo una tecnología de comunicación inalámbrica (Wifi).

El servidor principal se instaló en una Raspberry PI 3, la cual estuvo vinculada a un micro controlador llamado NodeMCU, este micro controlador fue programado con varios scripts, que enviaron las distintas peticiones a los sensores y actuadores.

El proyecto estuvo constituido por una interfaz gráfica amigable al usuario, el usuario logró acceder desde cualquier navegador web. Este sistema se inició con un Login de usuario, un sistema de iluminaria, sensor de movimiento, detector de gas y humo, control de persianas, y finalmente un sistema de auditoria que permitió visualizar los usuarios que lograron obtener acceso al sistema, cabe recalcar que también se diseñó y programó una interfaz de administración.

Como resultado se obtendrá una aplicación segura y accesible al usuario, con una interfaz administrativa que obtuvo el control de los dispositivos electrónicos.

Abstract

In the present project a domotic system of hardware and software was developed, which will allow to automate a house, with the purpose of facilitating the daily habits of the user, in such a way that only with manipulating a PC or a cell phone can have the control of your home, of course, this would depend on the services that are scheduled and installed within your home.

The objective is to have control of a home, through a wireless connection to manipulate electronic devices using mobile devices or a computer.

For the realization of the project, electronic devices such as microcontrollers, sensors, actuators that will send information to the main server were used. The server will process it and issue orders to the different electronic devices, through the use of HTTP GET and HTTP POST requests; it should be noted that the system developed will work under a wireless communication (Wifi) technology.

The main server will be installed in a Raspberry PI 3, which will be linked to a microcontroller called NodeMCU, this microcontroller will be programmed with several scripts, which will send the different requests to the sensors and actuators.

The project was provide a friendly user interface, where the user can access from any web browser. This system had a user login, a lighting system, motion sensor, gas and smoke detector, blind control, and finally an audit system that will allow users to see the system, it should be noted that it also provides an administration interface.

As a result, a secure and user-friendly application was obtained, with an administrative interface that will control electronic devices.

INTRODUCCIÓN

Antecedentes

En la actualidad, Los sistemas domóticos aportan múltiples beneficios y soluciones tanto para la vida cotidiana de las personas, como para el medio ambiente, entre estos se destacan seguridad, confort, comunicación y ahorro de energía (Universidad Mariano Gomez, 2016).

Problemática

En muchos hogares residenciales no existe automatización debido a que los costos de implementación son muy elevados. Los propietarios no muestran interés para satisfacer las necesidades de automatización de los hogares.

Para llevar a cabo estas tareas, los diversos dispositivos se encuentran interconectados por medio de redes interiores y exteriores de comunicación, cableadas o inalámbricas, con el objetivo de ser controlados dentro y fuera del hogar.

Justificación

Se desea desarrollar un sistema domótico integrando diferentes tecnologías que permitan dar facilidad a tareas básicas en residencias no automatizadas. Lo que se pretende con este proyecto es mejorar las condiciones de vida de la población en general transformando hogares o residencias simples en inteligentes.

La realización de este proyecto tiene un precio relativamente muy económico, con la finalidad de automatizar varias tareas. El software a utilizar es libre y puede implantarse en una Raspberry Pi, de tal modo que pueda clonarse y con unos pequeños ajustes de configuración se logre implementar en cualquier lugar con necesidades iguales o parecidas. La implementación de esta tecnología en los diferentes ámbitos

ayudara al ahorro de energía, a la eficacia de las tareas, a mejorar las condiciones de vida de personas con discapacidades físicas y la seguridad.

Objetivos

Objetivo General

Diseñar un sistema domótico que controle diferentes tareas básicas en residencias independientes con una red de sensores administrados por un servidor local doméstico y controlado por una interfaz operada desde un computador o Smartphone.

Objetivos Específicos

Investigar los tipos de sensores y actuadores que se utilizan para realizar sistemas domóticos.

Investigar y seleccionar el software adecuado para realizar las interfaces de control del circuito domótico.

Diseñar el diagrama electrónico con los sensores y actuadores que realizaran la tarea de automatización en el hogar.

Diseñar los diferentes scripts en la Raspberry pi para el respectivo control de los sensores y actuadores.

Realizar la conexión de la red de sensores con el módulo controlador para la gestión con la interfaz de los dispositivos controladores.

Realizar las respectivas pruebas de funcionamiento del sistema domótico.

Metodología

Método inductivo

Permitirá diseñar el sistema para que tenga acceso controlado, además de poder acceder al sistema de sensores interconectados. Para esto se construirá una interfaz de encendido de luces, sensores de movimiento, sensores de incendios y apertura de persianas controladas por el sistema embebido Raspberry pi.

Método Deductivo

Se realizará pruebas con los dispositivos electrónicos, de tal manera que se obtendrá un control adecuado del sistema y se reducirá considerablemente las fallas que se presenten en el proyecto.

Para el desarrollo del proyecto, se usó una metodología versátil que permita manejar procesos y hacerlos adaptables a las necesidades del proyecto. Por medio de hardware y software libre, se desea controlar sensores de movimiento HC-SR501, un detector de humo MQ-135, un servo motor SG90S, y por ultimo un Relay de 4 canales. Todos estos dispositivos serán controlados a través de un módulo Wifi NodeMCU; Para la interfaz de visualización, se utiliza un navegador web o dispositivo Android, en la que se muestra una notificación, cuando los sensores se activen.

CAPITULO 1

MARCO TEÓRICO

1.1. Domótica

La domótica, en términos generales se refiere a un sistema inteligente que permite integrar tecnología en edificios y viviendas con el fin de ejercer el control y sistematización, y así exista una comunicación entre el usuario y el sistema; es por ello que a la domótica también se la conoce como los sistemas de casas inteligentes (Rodriguez, 2014).

Se entiende por lo tanto que domótica es la unión de tecnologías que serán aplicables a las viviendas, brindando servicios como energía, seguridad, confort y comunicación. Todos los dispositivos que serán integrados en la vivienda podrán comunicarse por medio de redes cableadas o inalámbricas y para su control se lo podría hacer desde dentro y fuera del hogar.

1.2. Beneficios de la Domótica

En la domótica se pueden recalcar varios beneficios, los cuales están relacionados con el usuario en aspectos como su comodidad y seguridad. Entre los beneficios más importantes se mencionan algunos los cuales ayudan a los usuarios que deseen una residencia automatizada (Hernández Pérez, 2016).

1.2.1. Confort

Es uno de los aspectos que más sobresale en la domótica por lo cual las personas deciden adquirir esta tecnología, las mismas que dan al usuario facilidad de tener el control de diferentes áreas como sistemas de iluminación, climatización, y ventilación ya sea de manera local o a través de Internet (Jáimez, 2017).

1.2.2.Seguridad

Otro beneficio que destaca en la domótica es su gran aporte en la seguridad ya que dispone de dispositivos de movimiento, alarmas y video vigilancia que detectan de manera inmediata incendios, fugas de gas, escape de agua etc. Es decir, mantiene bajo control una vivienda a distancia asegurando que todo en ella funciona con normalidad (Jáimez, 2017).

1.2.3.Economía

Se considera el ahorro energético como un recurso eficiente dentro de la domótica ya que permite gestionar de modo inteligente la programación de los diferentes dispositivos incrementando la eficiencia energética y el ahorro (Jáimez, 2017) .

1.3. Componentes de un Sistema Domótico

La domótica consta de algunos componentes que se dividen en central de gestión, sensores y actuadores (Domótica Vento, 2017).

1.3.1. Central de Gestión

“Es el cerebro que se encarga de recolectar la información de los dispositivos, y a su vez procesa estos datos y genera las órdenes respectivas. Actualmente todas las funciones se encuentran embebidas en un solo dispositivo” (Torres Calvo, 2014).

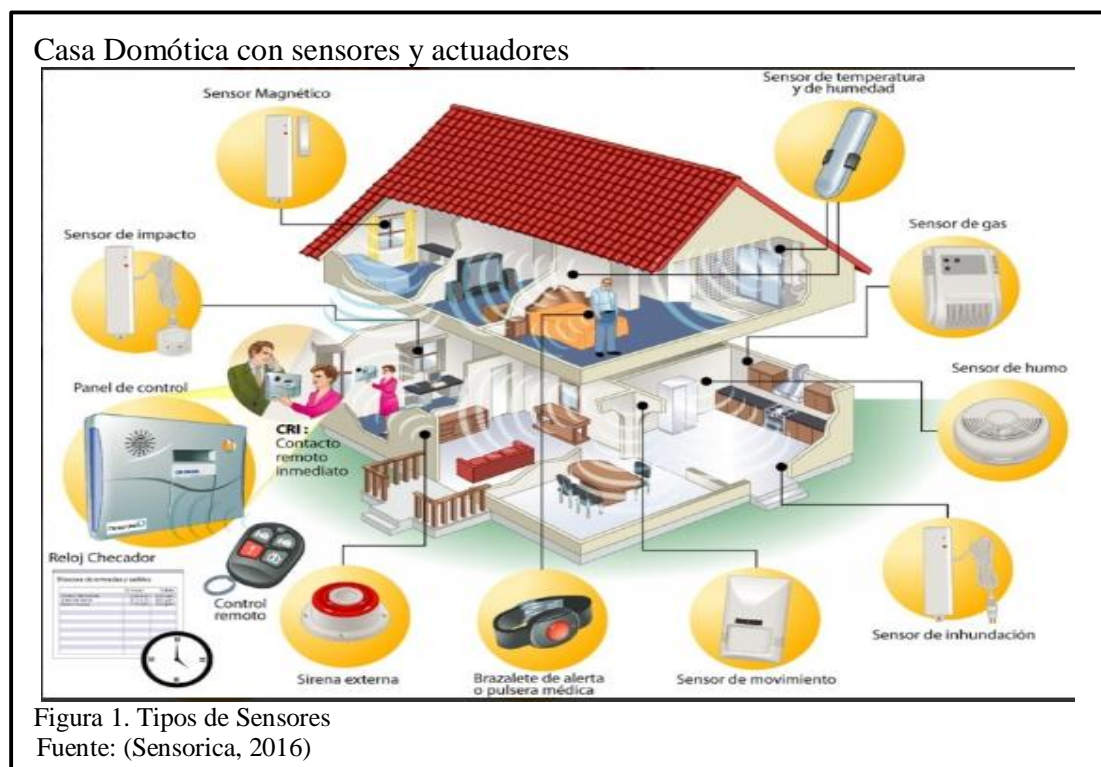
1.3.2. Sensores

“Son dispositivos que tienen la capacidad de convertir una magnitud física o química en una corriente eléctrica” (Blog de Domotica, 2015). Es decir, son capaces de detectar y transmitir cualquier información relevante de la vivienda. Se los puede clasificar en dos grupos diferentes unos son los sensores digitales y los otros son los sensores analógicos (Torres Calvo, 2014).

1.3.2.1. Tipos de Sensores

Existen diferentes tipos de sensores que son utilizados para traducir la información que llega del exterior en un impulso eléctrico. Este impulso usualmente pasa a una unidad de control en donde es analizado y transformado para generar una reacción o respuesta (Mejia Jervis, 2017).

Por lo que respecta a los tipos de sensores, existen sensores de contacto, ultrasonido, movimiento, deslizamiento, velocidad, corriente, temperatura, humedad, magnéticos, entre otros. En la Figura 1, se visualiza los distintos sensores y actuadores que pueden ser instalados en una residencia. Los nuevos avances tecnológicos facilitan que existan más sensores disponibles, con el fin de mejorar los procesos industriales y una infinidad de acciones que forman parte de la vida diaria (Parque de Innovación Tecnológica, 2015).

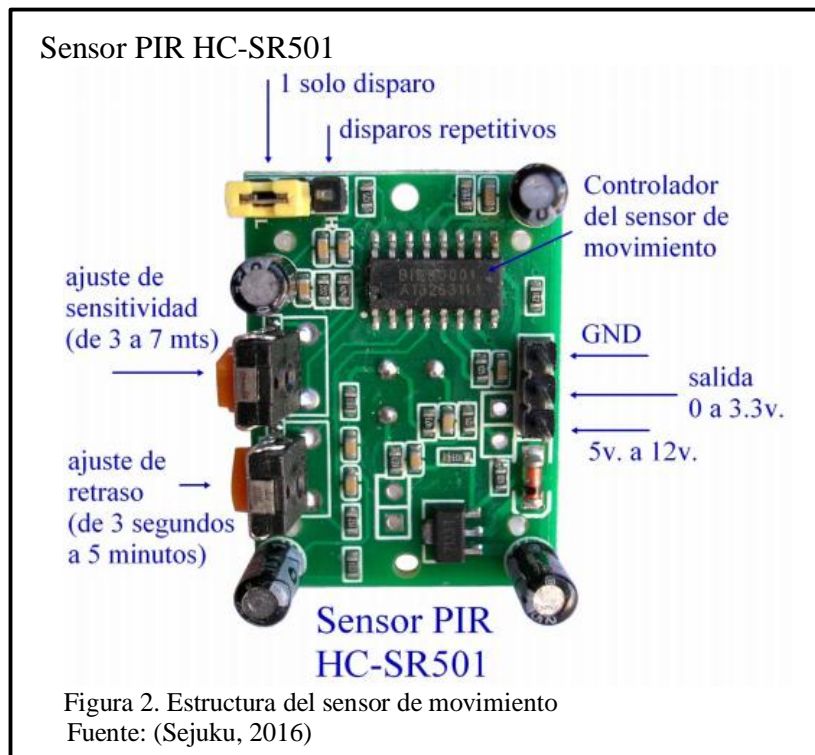


1.3.2.2. Sensor HC-SR501

Es un dispositivo eléctrico que mide la intensidad de calor, es decir, mide la radiación electromagnética que los seres vivos emiten. Como respuesta cambia su estado lógico PIN (0-1), con este estado se puede diferenciar si existe una presencia en el ambiente (Cuzco Carrion & Layana Vallejo, 2012).

Está compuesto por un lente especial (Lente de Fresnel) que alcanzan un rango de 6 metros de distancia y 19 zonas con un ángulo de apertura de 120°. Con estas especificaciones es prioritario que el sensor sea ubicado en una parte muy superior para que tenga mayor rango de alcance y detección (Jorge, 2017).

En la Figura 2, se visualiza la estructura del sensor PIR cuenta con 3 terminales, 2 para alimentación y uno de comunicación (detección de movimiento). También incorpora 2 potenciómetros y un jumper que permite modificar el comportamiento y adaptarlo a todo tipo de necesidades (Jorge, 2017).



Entre las características generales del sensor de movimiento se describen varias consideradas como principales. Estas hacen referencia al modo de funcionamiento adecuado en el sensor y son las siguientes (Bench, 2015):

- **Tiempo de retraso:** El ajuste del retardo de tiempo determina durante cuánto tiempo la salida del módulo del sensor PIR permanecerá alta después del movimiento de detección. El rango es de unos 3 segundos a cinco minutos. La siguiente ilustración muestra este ajuste.
- **Rango de sensibilidad:** Establece el rango de detección de 3 metros a 7 metros.

El modo de disparo le permite establecer el número de repeticiones en la detección de presencias. Los modos de disparo disponibles son los siguientes (Bench, 2015):

- **Disparos Simples:** el retardo de tiempo comienza cuando se detecta el movimiento por primera vez.
- **Disparos Repetidos:** cada movimiento detectado restablece el tiempo de retardo. Así que el tiempo de retardo comienza con el último movimiento detectado.

Las características del Sensor PIR se las detallan en la Tabla 1.

Tabla 1. Características del sensor PIR

Especificaciones	Notas
Sensor	piro eléctrico (infrarrojo)
Módulos	Sensor lente Fresnel, Controlador PIR BISS0001, regulador de disparos.
Rangos de visión	3 m a 7 m
Visión de ángulo	< 100°
Salida pulso lógico	3.3v
Consumo de corriente	< 50 Ma
Voltaje	4.5 a 20 VDC

Fuente: (NaylampMechatronics, 2018)

1.3.2.3. Modelo MQ-135

El sensor MQ135 tiene una capa de óxido de estaño recubierta por una malla de gasa de acero inoxidable. MQ-135 permite detectar diferentes tipos de gases contaminantes en el ambiente y puede ser calibrado para obtener medidas de dióxido de carbono en el aire (Arciniega, 2017).

También permite ajustar la sensibilidad hacia los gases en el aire de acuerdo a cada sustancia, como la mayoría de sensores MQ tiene un tamaño pequeño, instalación rápida y fácil de usar. En la Figura 3, se visualiza el módulo MQ el cual, en su estructura consta de 4 pines, dos para alimentación de 5V y dos pines para la salida analógica y un pin la salida digital. Este sensor puede ser utilizado para tomar lecturas sobre la calidad del aire, dentro de oficinas, hogares y lugares donde se requiera detectar los gases antes mencionados (Ganchoso, 2018).

Módulo MQ-135

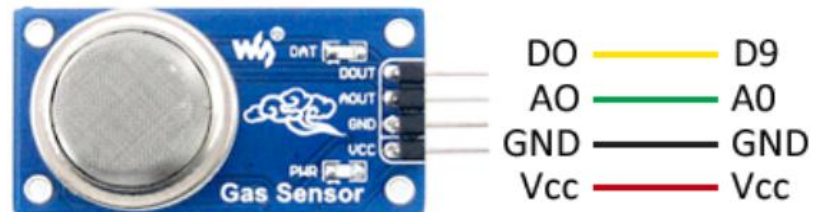


Figura 3. Sensor de Humo con salida digital y salida analógica
Fuente: (Llamas, 2016)

En la Figura 4, se describen las características más importantes del sensor MQ135.

Características del sensor MQ135

Parámetro	Descripción
Voltaje operativo.	De 2.5V a 5V
Corriente de operación.	160mA
Disipación de potencia.	<900mW
Temperatura de operación.	-10°C a 45°C
Humedad de operación.	<65%
Temperatura de almacenamiento.	-20°C a 70°C
Rango de medición.	10-10000ppm de CO2
Margen de error	±100ppm
Tiempo de respuesta.	<10 segundos
Estabilidad.	No calculada
Fiabilidad.	Alta

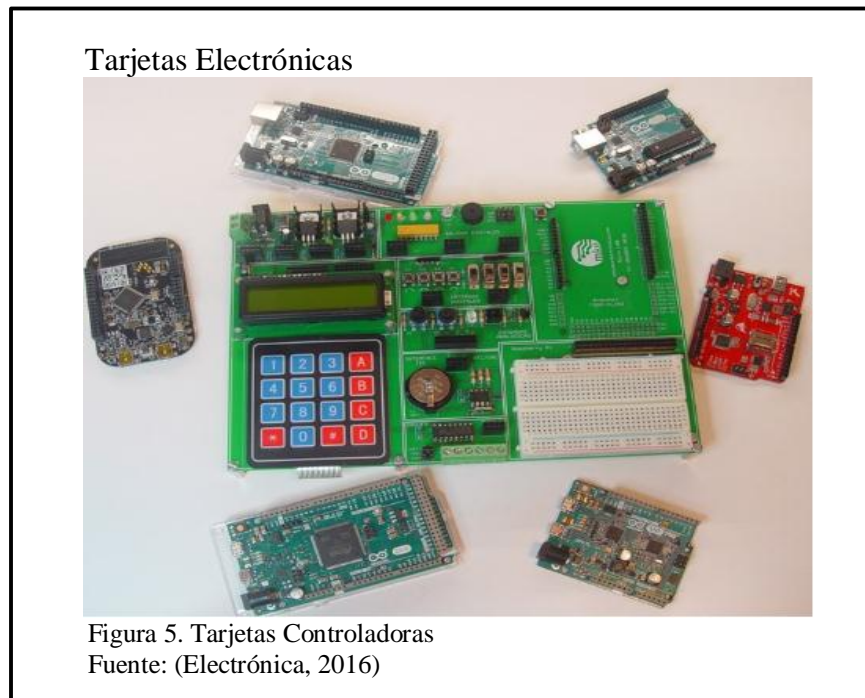
Figura 4. Características del sensor MQ135
Fuente: (Banggood, 2016)

1.3.3. Controladores

Son llamadas tarjetas controladoras o electrónicas por encargarse de procesar la información, trabajan como controladores dentro del sistema domotico procesando las señales ya sean analógicos o digitales. Esto dependería del tipo de sensor propuesto (Rodríguez & Quito, 2017).

La función de los controladores es encargarse de receptar señales leídas por los sensores, para luego procesarlas. Dependiendo de la programación desarrollada en la

tarjeta electrónica se emitirá las órdenes correspondientes a los demás dispositivos (Rodríguez & Quito, 2017). En la Figura 5, se hace referencia a las tarjetas controladoras más conocidas para realizar tareas y control domótico.



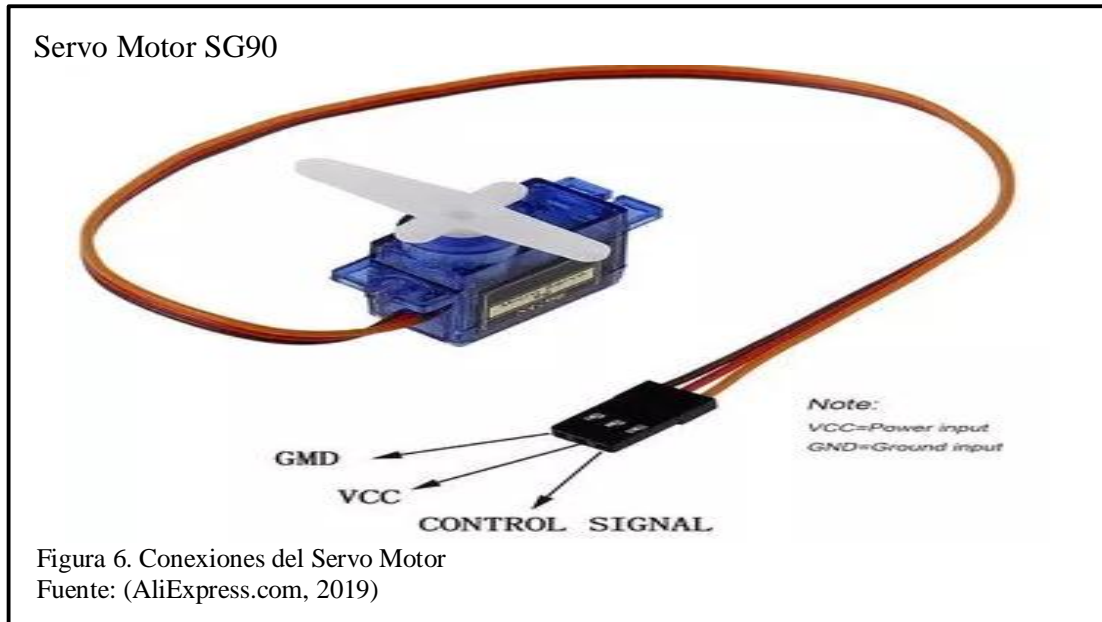
1.3.4. Actuadores

“Un actuador es el encargado de actuar sobre una operación iniciada por el controlador, es decir transforma una acción de encendido o apagado” (Mecaxis.Cat, 2007). Con un enfoque más claro los actuadores toman decisiones y ejecutan las acciones apropiadas sobre el entorno. En la mayoría de los casos estos dispositivos (sensor y/o actuador) funcionan conjuntamente, es decir, que permiten administrar su estado conjuntamente (Torres Calvo, 2014).

1.3.4.1. Actuador Micro Servo SG90

El Servo Tower Pro SG90 es un actuador diminuto y liviano, puede girar entre los 180 grados (90 en cada dirección) (Hernández Pérez, 2016).

Este Servo es ideal para sistemas de baja potencia que requieran de piezas en movimiento, permite mantener una posición dentro del rango de operación del dispositivo. Por otro lado, permite controlar la velocidad de giro (Hernández, 2016).



En la Figura 6, se muestra la distribución de los cables el cual se describe desde el de color Café como GND (tierra), Rojo como VCC (5V) y Naranja como Señal de control (PWM). En la Figura 7, se muestran las características del actuador Tower Pro SG90 (Smelpro, 2016).

Características del Servo Motor

Modulación	Analógico
Torque	4.8v/1.8 kg-cm
Velocidad	0.1 <u>sec</u> /60°
Ancho pulso	500-2400 <u>us</u>
Peso	9.0g
Temperatura	0°C – 55 °C
Tipo de Material	Plástico

Figura 7. Características del actuador Tower ProSG90
Fuente: (Workship Magazine, 2017)

Ideal para trabajar con diversidad de plataformas de desarrollo como Arduino, Raspberry Pi, micro controlador. Puede ser alimentado por USB, pero se recomienda

alimentar por separado el micro controlador y los servos, ya que el ruido eléctrico puede dar lugar a errores en la ejecución del programa (Smelpro, 2016).

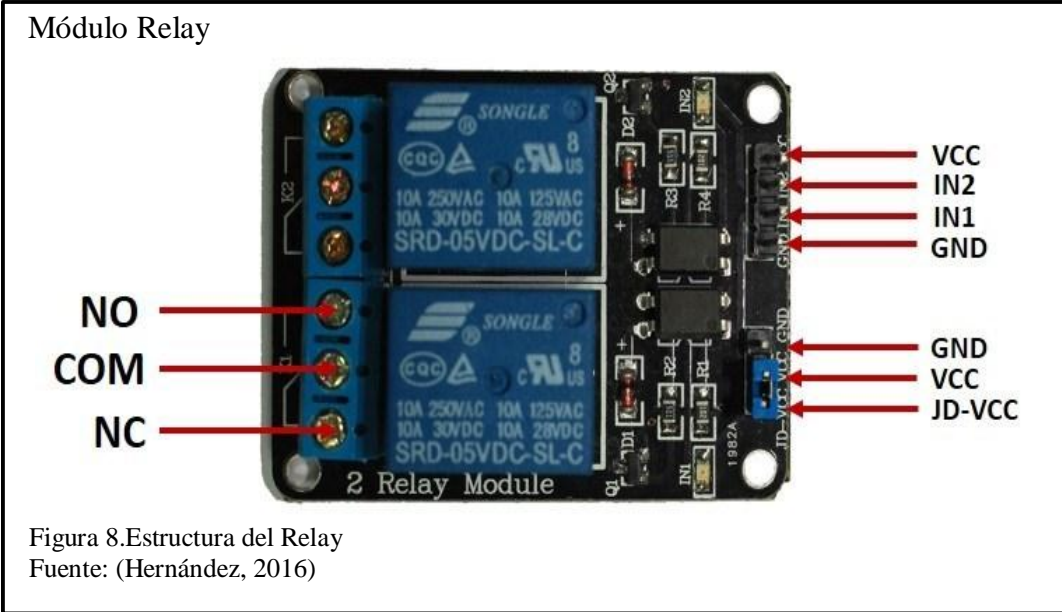
Cabe recalcar que existen varios servomotores que tienen distintas funciones entre estos tenemos los siguientes:

- **Servomotor de giro limitado:** Este servo es débil, y algo lento. Solo puede moverse hacia adelante y hacia atrás 180 grados (Barela, 2018).
- **Servomotores de rotación continua:** Son básicamente servos normales mutilados que técnicamente no merecen ser llamados servos porque carecen del control de retroalimentación que es el sello distintivo de cualquier sistema servo (Malášek, 2011).

1.3.4.2. Módulo Relay

Es un interruptor que funciona por conmutación, es decir sus módulos de soporte permiten el control de un interruptor eléctrico que se puede encender o apagar, es decir dejar pasar la corriente o no. Se lo puede controlar mediante voltajes bajos, como los 5V. (Santos, 2019).

Este módulo está diseñado para ser controlado directamente por una amplia gama de micro controladores como Arduino, PIC, ARM etc. En la Figura 8, se observa los componentes que realizan las operaciones adecuadas para el funcionamiento correcto del Módulo Relay.



Las características del Relay se detallan en la Tabla 2.

Tabla 2. Características Relay

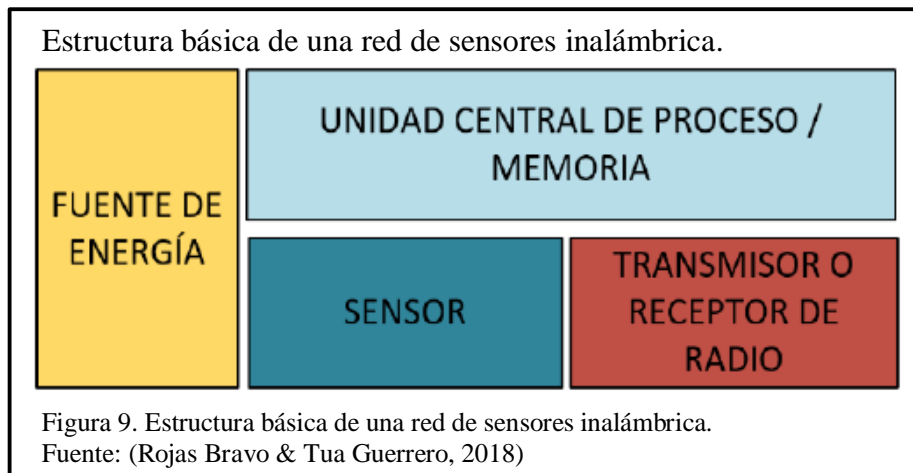
Especificaciones	Notas
Señal de Control	3.3V o 5V TTL
Número de Canales	1 CH, 2 CH, 3CH, 4CH
Capacidad máxima corriente	10A/250VAC, 10A/30VDC
Tiempo de acción	10 ms / 5 ms
Voltaje de Operación	5VDC

Fuente: (Hernández Pérez, 2016)

1.4. Red de Sensores Inalámbricos

Son redes que sirven para implementar y recopilar información de los sensores inalámbricos. El uso inalámbrico elimina la necesidad de cableado y ahorra en los costos de cableado e instalación. Además, se puede construir al hacer de una red de sensores inalámbrica que no se pudo realizar debido a la restricción del sitio de instalación y el costo (Mono Wireless, 2015).

La conexión inalámbrica nos permite saber el estado de un dispositivo lejano y a su vez controlarlo. Sin embargo los datos de medición y la ubicación de la red de sensores inalámbricos (WSN) son muy amplios y variados.



La red de sensores contiene una fuente de alimentación que se encarga de proveer energía al circuito, y está compuesto por un módulo transmisor/emisor para el envío y recepción de datos tomados de los sensores (ver Fig. 9) (Rojas Bravo & Tua Guerrero, 2018).

1.5. Tipos de Arquitectura

Para una topología de red de sensores se deberá seleccionar la más adecuada, entre las principales se encuentran las siguientes topologías (Rodríguez J. A., 2017):

- **Topología Centralizada:** es el encargado de recibir los datos de los diferentes dispositivos, para procesarlos y gestionarlos, de manera que se envíen las órdenes adecuadas a cada actuador (ver Fig. 10).

Arquitectura Centralizada

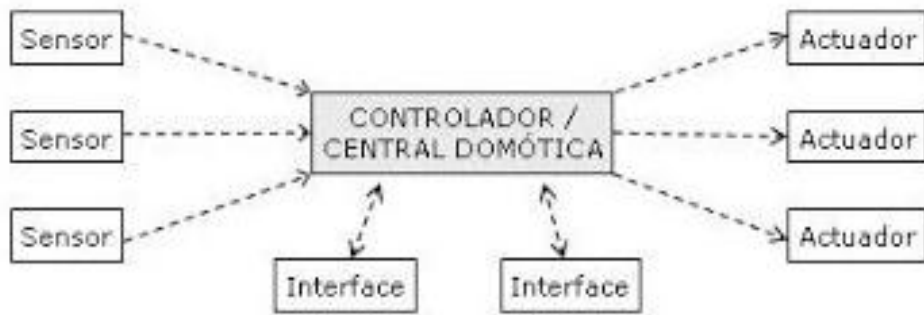


Figura 10. Arquitectura Domótica Centralizada

Fuente: (Arqcompus-domotica, 2009)

- **Topología Distribuida:** nos permite que los dispositivos funcionen por sí mismos ya que contienen un controlador propio (ver Fig. 11).

Arquitectura Distribuida

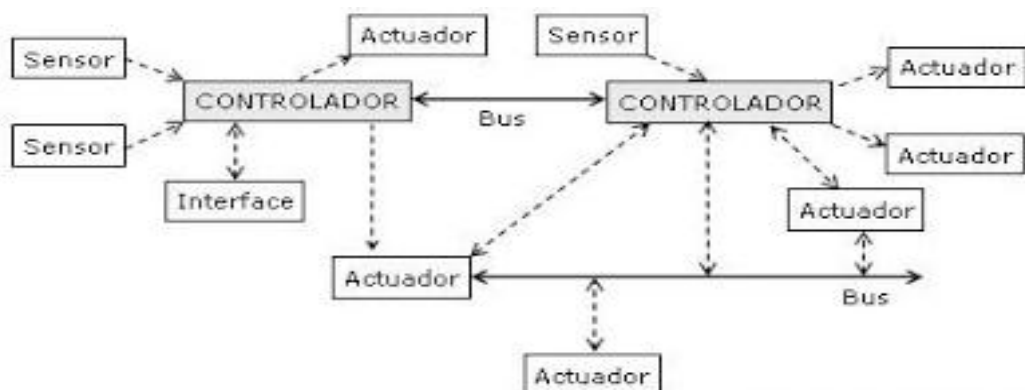
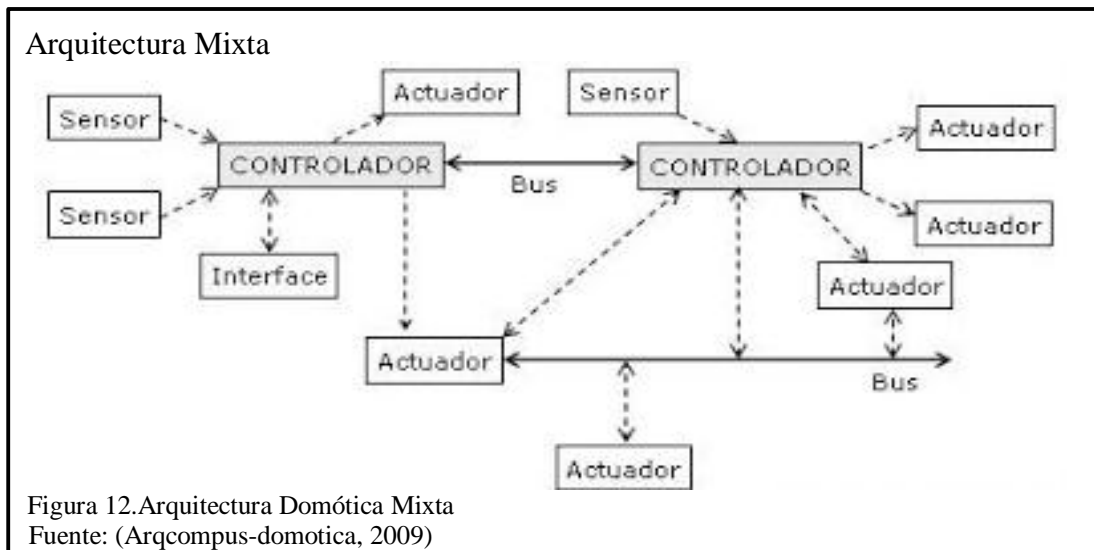


Figura 11. Arquitectura Domótica Distribuida

Fuente: (Arqcompus-domotica, 2009).

- **Topología Mixta:** Se puede mencionar que es la combinación entre la topología centralizada y distribuida (ver Fig.12). Es decir, que pueden trabajar como controladores y procesadores de información para así transmitirla a otros dispositivos sin necesidad de pasar por un controlador para ello (Rodríguez J. A., 2017).



1.5.1.Medios de Interconexión

Los medios de interconexión en un sistema domótico pueden ser: alámbricos o inalámbricos, dependiendo si se encuentran cableados a la central o no. En función de la característica de los medios de interconexión, se establecen distintos tipos de centrales (Eufemio, 2008):

- **Centrales cableadas:** todos los dispositivos (sensores y actuadores) están conectados por cables a la central, lo cual permite el control de todo el sistema. En el caso que existiera una falla eléctrica no lo afectaría, ya que estos sistemas poseen una batería de respaldo que beneficia a todos sus sensores y actuadores si se produce un fallo del suministro eléctrico.
- **Centrales inalámbricas:** Poseen una fuente eléctrica de 5V.

1.5.1.1. Comunicación Inalámbrica Wifi

Wi-Fi es un estándar inalámbrico, que permite la interconexión entre dispositivos mediante una LAN inalámbrica. En otras palabras, todo dispositivo que cuente con este estándar debería conectarse a una red inalámbrica. Wi-Fi utiliza el estándar internacional IEEE 802.11, su banda de frecuencia y velocidad de comunicación están determinadas por el estándar (Eufemio, 2008).

1.6. Sistemas Embebidos

Sistemas embebidos se refiere a todo circuito electrónico digital, que permiten trabajar a tiempo real y a su vez cumplir con una tarea específica. Los sistemas embebidos se componen de microprocesadores que permiten al software ser productivo y beneficioso en varios ambientes como ejemplo hoy en la actualidad los autos modernos, en teléfonos móviles, refrigeradoras, alarmas, equipos médicos entre otros. Muchos sistemas embebidos no están equipados con programas complicados y, desde el punto de vista del costo, la computadora utilizada tiene solo el rendimiento mínimo requerido para controlar los dispositivos, tanto en la capacidad de procesamiento como en la capacidad de almacenamiento (Salas Arriarán, 2015).

1.6.1. Raspberry Pi 3

Raspberry Pi es una pequeña computadora desarrollada por la Raspberry Pi en el Reino Unido. Aunque es del tamaño de una tarjeta, está equipado con puertos de conexión HDMI, USB y LAN por cable, por lo que tiene una excelente escalabilidad (Workshop Magazine, 2017).

Lo más novedoso de la Raspberry Pi 3 fue su conectividad, ya que se integró Wifi y de Bluetooth en la misma placa, por lo que, no se utilizaría conectores USB (Garcia, 2016).

Esta placa es capaz de operar con sistemas basados en Linux, y son muy útiles para desarrollar trabajos electrónicos y de domótica. Este dispositivo no tiene integrado discs de almacenamiento ya que trabaja con una tarjeta micro SD donde se realiza la instalación del SO (Mera, 2018).

En la Figura 13, se detallan las características de funcionamiento de la Raspberry Pi.

Características de la Raspberry Pi 3	
Componente	Característica
Procesador	- Chipset Broadcom BCM2387. - 1,2 GHz de cuatro núcleos ARM Cortex-A53
GPU	- Dual Core VideoCore IV® Multimedia Co-procesador. Proporciona Open GL ES 2.0, OpenVG acelerado por hardware, y 1080p30 H.264 de alto perfil de decodificación. - Capaz de 1 Gpixel/ s, 1.5 Gtexel/ s o 24 GFLOPs con el filtrado de texturas y la infraestructura DMA.
RAM	1GB LPDDR2
Conectividad	- Ethernet socket Ethernet 10/100 BaseT - 802.11 b/ g/ n LAN inalámbrica y Bluetooth4.1 (Classic Bluetooth LE) - Salida de vídeo: * HDMI rev 1.3 y 1.4 * RCA compuesto (PAL y NTSC) - Salida de audio* jack de 3,5 mm de salida de audio, HDMI - USB 4 x Conector USB 2.0 - Conector GPIO * 40-clavijas de 2,54 mm (100 milésimas de pulgada) de expansión: 2x20 tira * Proporcionar 27 pines GPIO, así como 3,3 V, +5 V y GND líneas de suministro. - Conector de la cámara de 15 pines cámara MIPI interfaz en serie (CSI-2) - Pantalla de visualización Conector de la interfaz de serie (DSI), Conector de 15 vías plana flex cable con dos carriles de datos y un carril de reloj - Ranura de tarjeta de memoria Empuje / tire Micro SDIO

Figura 13. Características de la Raspberry Pi 3
Fuente: (Garcia, 2016)

El Raspberry se conecta a una TV o Pantalla LCD con entrada HDMI. Es un pequeño computador muy barato, donde es posible construir un servidor de archivos estable de gran capacidad. El cual permitiría el intercambio de archivos dentro de una red local (Kawashima, 2019).

1.6.1.1. Modelos de un Raspberry PI

Existen dos modelos de Raspberry PI 3 que están en auge actualmente en el mercado, y estos dos son el modelo B y el modelo B+.

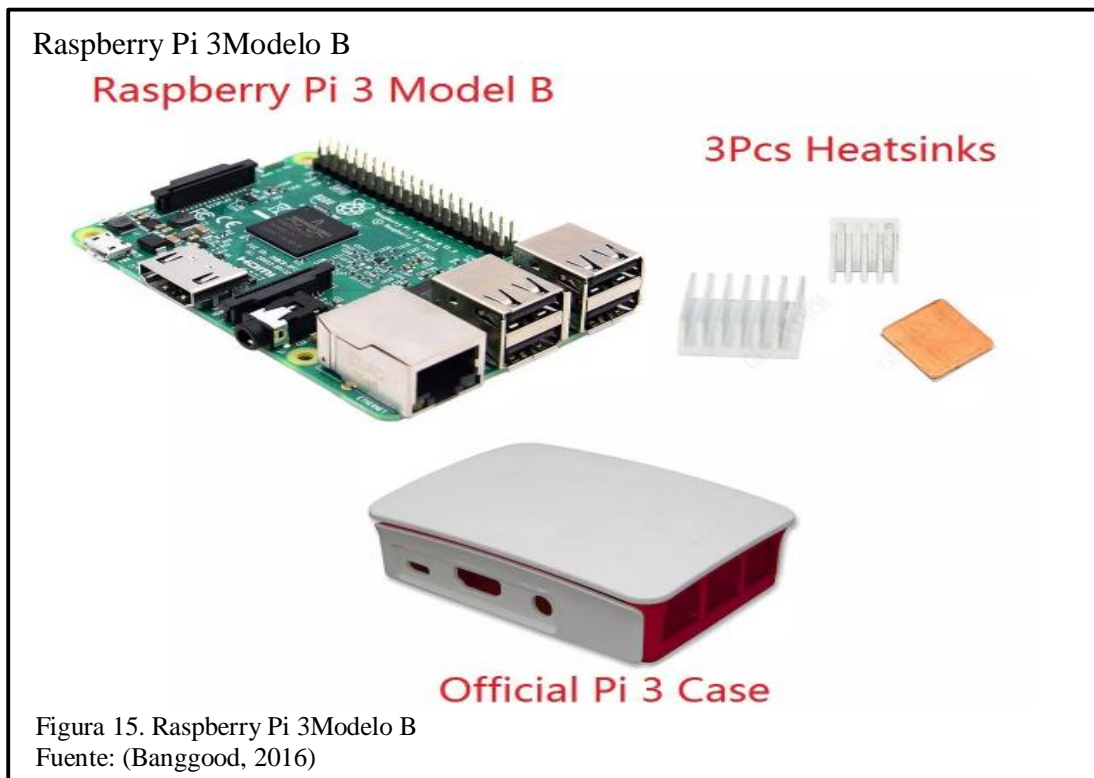
Características Raspberry PI 3 B y B+		
Especificaciones	Raspberry PI 3 MODEL B+ (2018)	Raspberry PI 3 MODEL B (2016)
PROCESADOR	Broadcom BCM2837B0, Cortex- A53 (ARMv8) 64-bit SoC @ 1.4GHz	Broadcom BCM2837, Cortex-A53 (ARMv8) 64-bit SoC @ 1.2GHz
RAM	1GB RAM	1GB RAM
CONECTIVIDAD	WiFi 802.11.b/g/n/ac de doble banda 2.4GHz y 5GHz	WiFi 802.11 b/g/n (2.4GHz)
	Bluetooth 4.2	Bluetooth 4.1
	Puerto Ethernet de hasta 300Mbps	Puerto Ethernet de hasta 100Mbps

Figura 14. Características Raspberry PI 3 B y B+
Fuente: (Xataka, 2018)

Se observa en la figura 14, que se realizan varias modificaciones en su tecnología lo que le hace una versión mejorada. Es decir es mucho más rápido en procesamiento y en conectividad, se mantendrá los mismos puertos y el mismo tamaño que su predecesora.

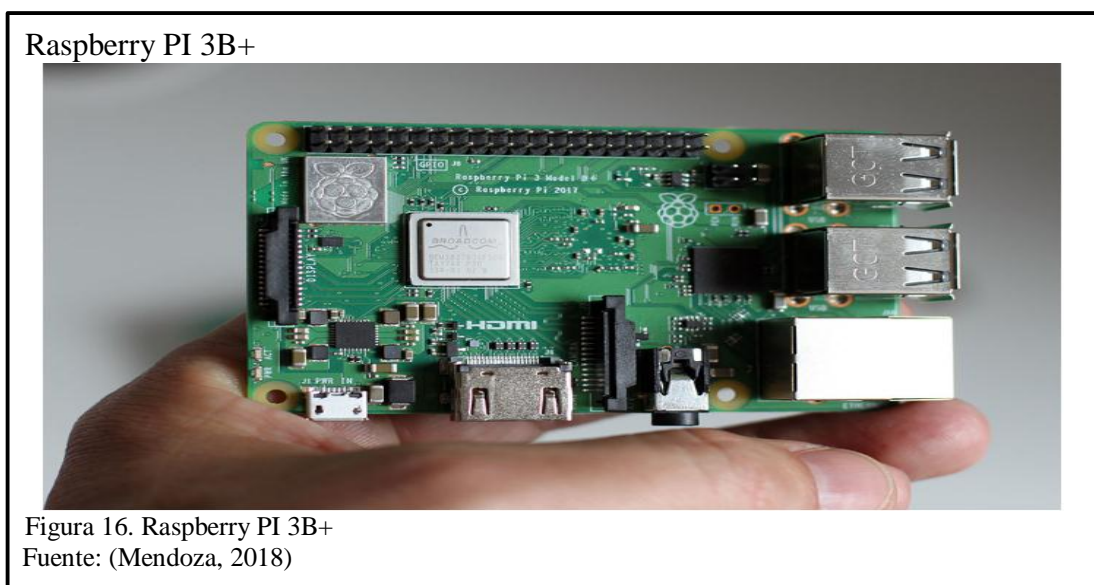
1.6.1.2. Raspberry Pi 3 Modelo B

Entre las novedades más destacadas de la Raspberry Pi 3B tiene un nuevo núcleo de 64 bits y WI-FI interno y Bluetooth. Su alto rendimiento mejoró diez veces en comparación con la generación anterior, es decir, debe equiparlo con un sistema de enfriamiento fuerte (ver Fig. 15). El conjunto de disipadores de calor puede satisfacer sus demandas muy bien. La cinta adhesiva termo conductora de seda térmica es de la empresa 3M Modelo 8810, puede soportar altas temperaturas de tiempo alto hasta 100 °C (Banggood, 2016).



1.6.1.3. Raspberry Pi 3 Modelo B+

Este modelo cuenta con los mismos puertos de conexión, y el mismo tamaño que su predecesora (ver Fig. 16). La única diferencia es la versión (4.1 a 4.2) del Bluetooth que ha sido actualizada (Mendoza, 2018).



Desde la Raspberry PI 3B+ se puede acceder a Internet e incorporar equipos para construir sistemas domóticos. Debido a que tiene varios terminales, tiene una gran versatilidad para realizarlos. Por ejemplo, la información de múltiples dispositivos / sensores se puede recopilar y enviar a dispositivos (Rodríguez, 2014).

1.6.2. Puertos GPIO

La Raspberry Pi contiene un procesador llamado BCM2837 que es fundamental para el procesamiento. Este procesador tiene una gran cantidad de pines. Hay 53 pines GPIO para este procesador, 26 de los pines están conectados directamente a los pines GPIO de la Raspberry Pi (Laboratorio de herramientas, 2015).

Los puertos GPIO se deberían configurar como entrada o salida, pero se debe tener en cuenta que los puertos son completamente digitales, y por lo tanto, no permite leer señales analógicas. Estos puertos GPIO se puede usar para enviar y recibir señales de Raspberry Pi hacia el exterior (Harrington, 2015).

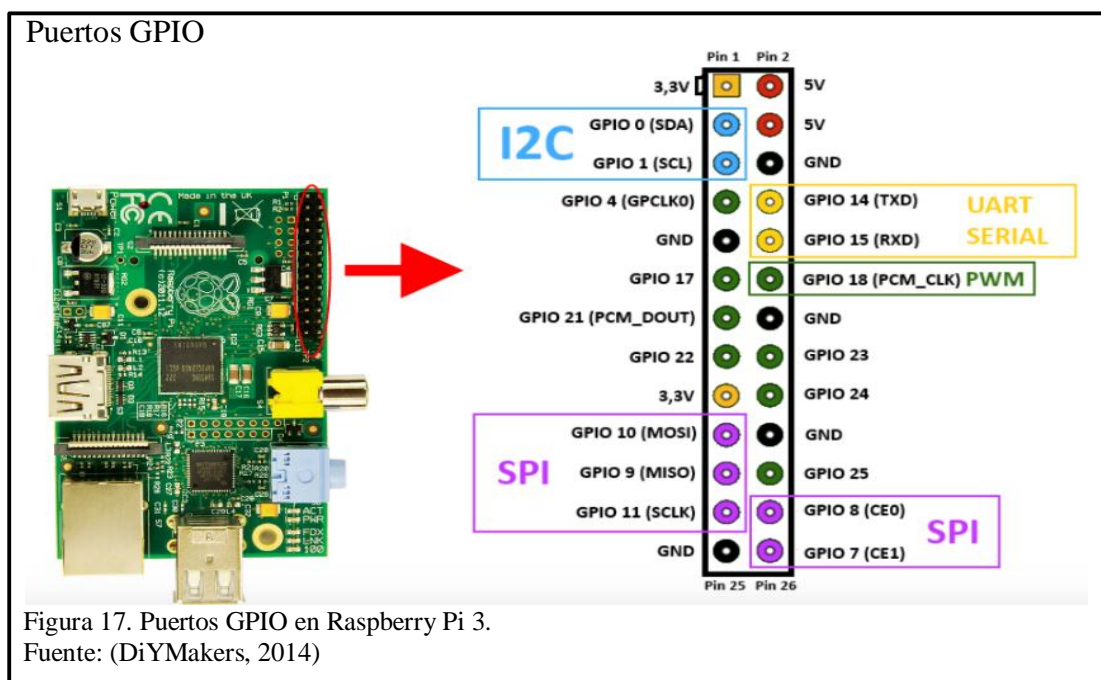
El puerto GPIO normalmente se reserva para micro controladores, consta de varios pines que se pueden utilizar como entradas o salidas, la corriente consumida y generada por estos pines es de 3.3V y no deben sobrepasarse de los 5V. (Cabeza, 2018).

Cabe recalcar que no todos los pines del puerto GPIO son de entrada y salida, también existen varios puertos que cumplen distintas funciones (Cahuna & Pari, 2017):

- **Voltajes:** Tiene dos pines de 5V y dos pines de 3V que se encuentran en la placa, así como unos pines de conexión a tierra (0V) que no se pueden configurar. Los pines restantes son pines de propósito general, lo que significa que las salidas y las entradas son tolerantes a voltajes de 3V.

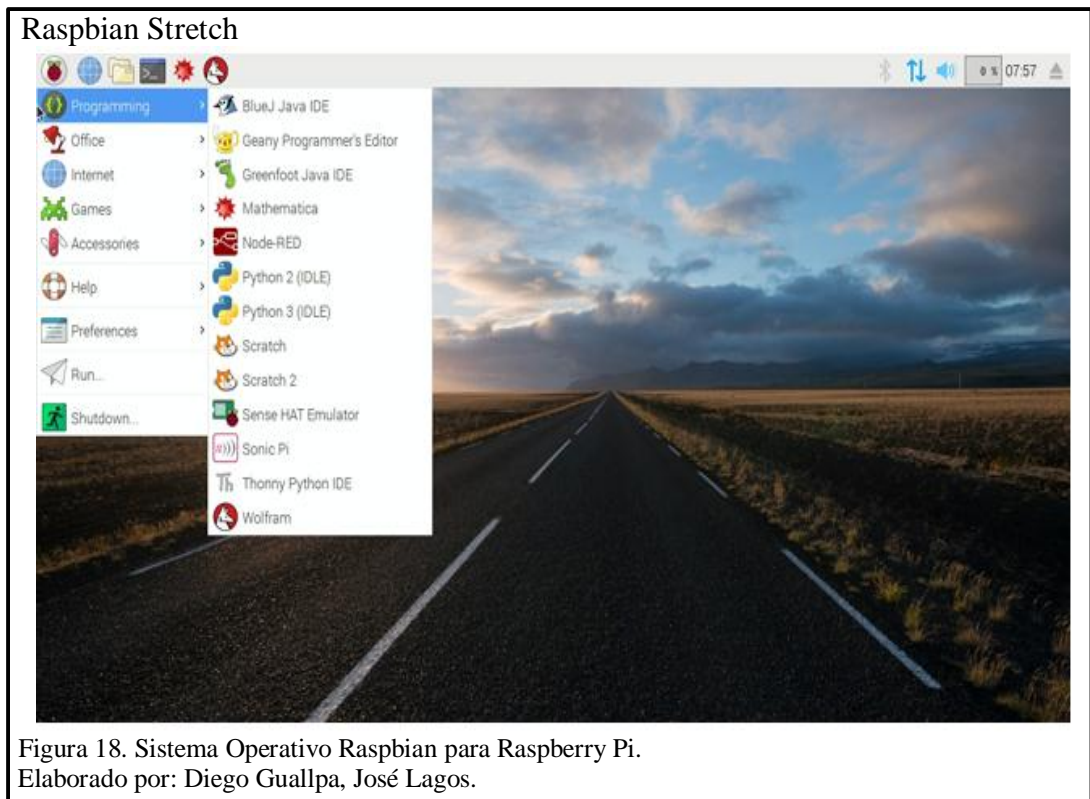
- **Voltajes GND:** Son pines que por el momento no tienen ninguna función en las placas actuales y se encuentran señalados como GND.
- **Pines de entrada y salida:** Estos pines tienen múltiples funciones y vienen marcados desde el 2 al 26, se los configuran a través de software.
- **Interfaz UART:** UART (transmisor de receptor asíncrono universal) utilizado para interconectar sensores y otros dispositivos. A este grupo de pines se los puede encontrar en la placa como TXD y RXD que sirven para comunicaciones en serie, buses de comunicación, etc. (Garcia, 2016).

En la figura 17, se muestra un esquema de los pines de la Raspberry Pi 3.



1.6.3. Sistema Operativo Raspbian

Raspbian usa una distribución de Linux llamada Debian, modificada y optimizada para sacar el máximo partido a este micro-computador (ver Fig. 18). Raspbian se utiliza igual que Debian, es muy sencilla, estable, rápida y se puede realizar cualquier tipo de instalaciones (Rubén, 2019).



Raspbian es recomendado para Raspberry Pi (al estar optimizado para su hardware), ya que utiliza un entorno llamado PIXEL como escritorio y Midori como navegador web (Abellán, 2019).

Este sistema operativo servirá como servidor web para la comunicación entre la página web y los dispositivos (sensores y actuadores), viene integrado varias herramientas de programación que se las puede usar para la codificación de los dispositivos (Arévalo, 2017).

1.7. Arquitectura de un Sistema Embebido

Un sistema embebido se compone por un microprocesador y un software que se ejecute sobre éste, pero este software necesita un lugar donde poder guardarse para luego ser ejecutado por el procesador. Esto se llama memoria RAM o ROM (Nava Oscar, 2016).

Todo sistema embebido necesita una cierta cantidad de memoria, que incluso se encuentran dentro del mismo chip del procesador. Adicionalmente cuentan con una serie de salidas y entradas necesarias para comunicarse con dispositivos exteriores (Nava Oscar, 2016).

En su memoria sólo reside el programa destinado a gobernar una aplicación concreta. Sus líneas de entrada/salida (I/O) soportan el conexionado de los sensores y actuadores del dispositivo a controlar y todos los recursos complementarios disponibles tienen como propósito atender a sus requerimientos. Estas son las características más importantes que tienen comúnmente los sistemas embebidos (Nava Oscar, 2016).

1.7.1. NodeMCU

Es una plataforma de código abierto que se lo utiliza para el desarrollo de proyectos que se pueden conectar a Internet por medio de Wifi, en la actualidad existe una gran cantidad de software compatible y librerías para la programación del NodeMCU-ESP8266 (ver Fig. 19). La gran ventaja de este desarrollo es su programación que lo hace totalmente transparente ya que dispone de una conexión USB para la comunicación directamente al computador (GitHub,Inc , 2016).

Además cuenta con la facilidad de acceder a sus pines GPIO, viene integrado una memoria Flash de 1Mb para el almacenamiento y 2 puertos RX (recepción)/TX (envío), en definitiva, es muy económico para el desarrollo de aplicaciones de IoT ya que contiene el protocolo de conexión inalámbrica Wi-Fi (GitHub,Inc , 2016).

NodeMCU

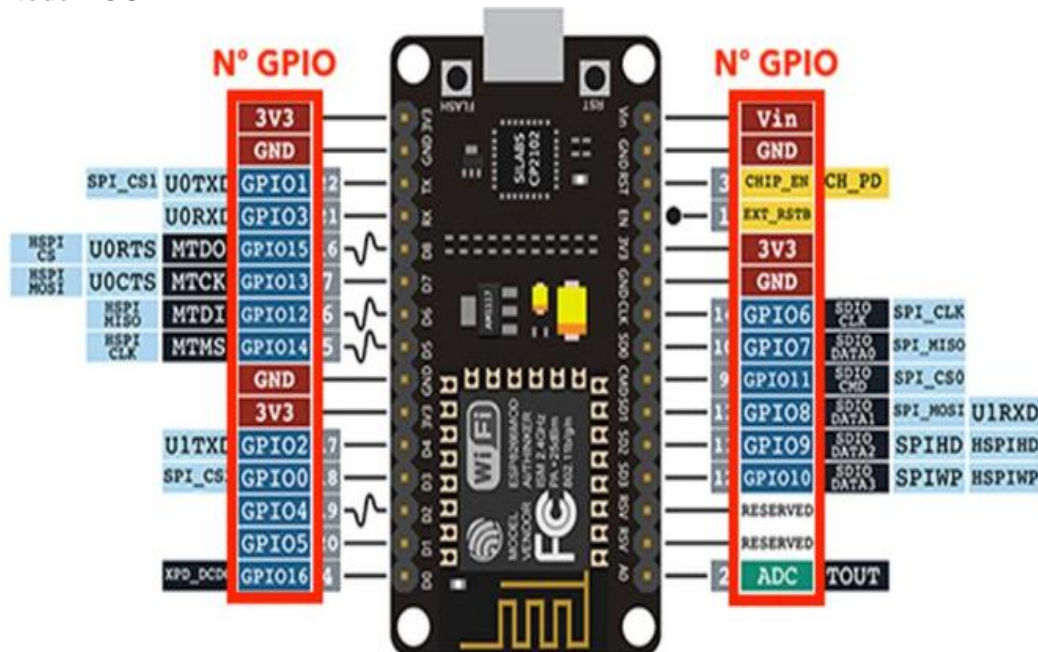


Figura 19. Pinout de una placa NodeMCU.
Fuente: (Koyanagi, 2018)

A continuación en la Tabla 3, se muestra las especificaciones del NodeMCU ESP8266.

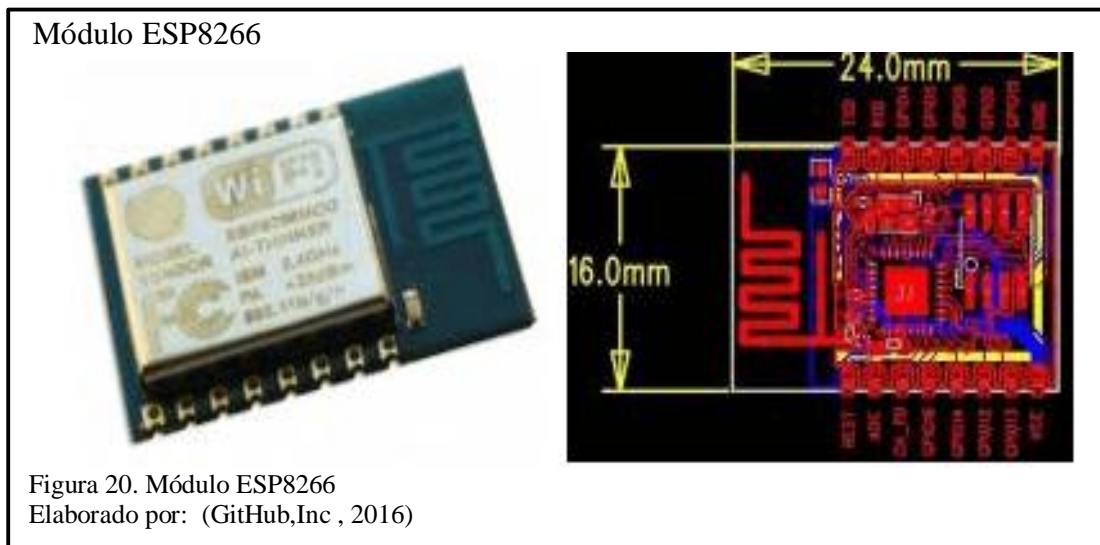
Tabla 3. Especificaciones del NodeMCU ESP8266

Especificación	Notas
Hardware	Controlador ESP8266
Conexión a internet	Wifi
Protocolo inalámbrico	802.11b/g/n
Consumo de corriente al utilizar	100-110mA

Fuente: (Rodrigo, 2016)

1.7.2. Módulo Wifi ESP8266

ESP8266 es el nombre de un micro controlador diseñado por una compañía china llamada Espressif Systems en su sede en Shanghái (ver Fig. 20). Pero su producción hasta principios del año 2014, donde se anunció que este chip sería una excelente solución automática de redes WiFi ya que tiene la capacidad de ejecutar aplicaciones independientes (Ceja José, 2017).



El módulo ESP8266 ofrece una solución de red Wi-Fi completa y autónoma, que le permite alojar la aplicación o descargar todas las funciones de red Wi-Fi de otro procesador de aplicaciones, Tiene caché integrado para mejorar el rendimiento del sistema en dichas aplicaciones y para minimizar los requisitos de memoria (Marian, 2015).

En la figura 21, se muestran las características generales:

EspecificacionesESP8266	
Voltaje	3.3 V
Consumo de corriente	10 μ A – 170 mA
Memoria Flash	16 MB máx. (512 k normal)
Procesador	Tensilica L106 32 bit
Velocidad del procesador	80 – 160 MHz
GPIOs	17
Analógico ha digital	1 entrada con 10 bit de resolución (1024 valores)
Soporte de 802.11	b/g/n/d/e
Máximas conexiones simultáneas	5

Figura 21. Especificaciones del Micro controlador ESP8266
Fuente: (Ceja José, 2017)

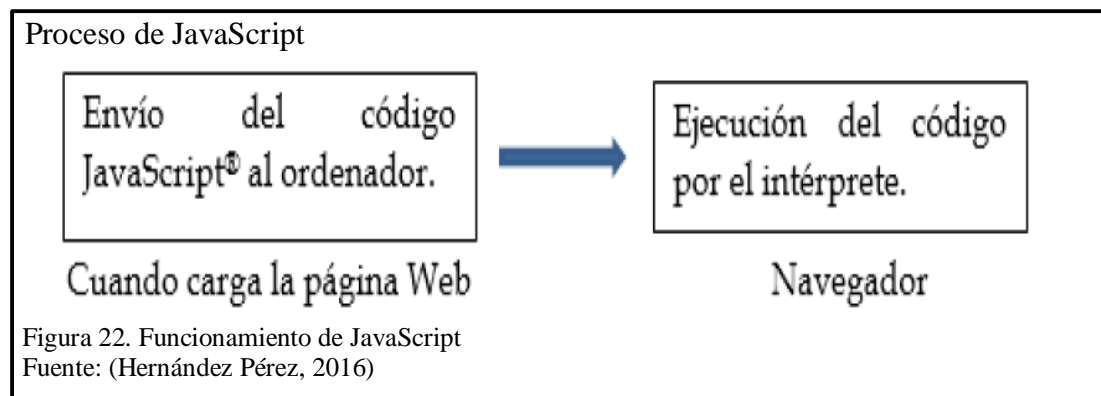
1.8. Lenguajes de Programación Web

A medida que Internet of Things (IoT) continúa evolucionando, puede ser difícil rastrear qué herramientas son las más populares para diferentes propósitos. Entre algunos de los Lenguajes de programación podemos mencionar los siguientes:

- **HTML:** HTML (Hyper Text Markup Language) es un lenguaje desarrollado para crear páginas web. Actualmente, la mayoría de las páginas web publicadas en Internet están creadas en HTML, estas páginas cuentan con hipervínculos que permiten movernos de una página a otra (Sejuku, 2016).
- **CSS:** CSS es un mecanismo que permite configurar los parámetros de una página web, entre estos parámetros se encuentra el color, tamaño, sangrías en el documento, tipo de letra etc. CSS puede trabajar en conjunto con HTML con

el propósito de dar forma a la estructura de la página web (Hernández Pérez, 2016).

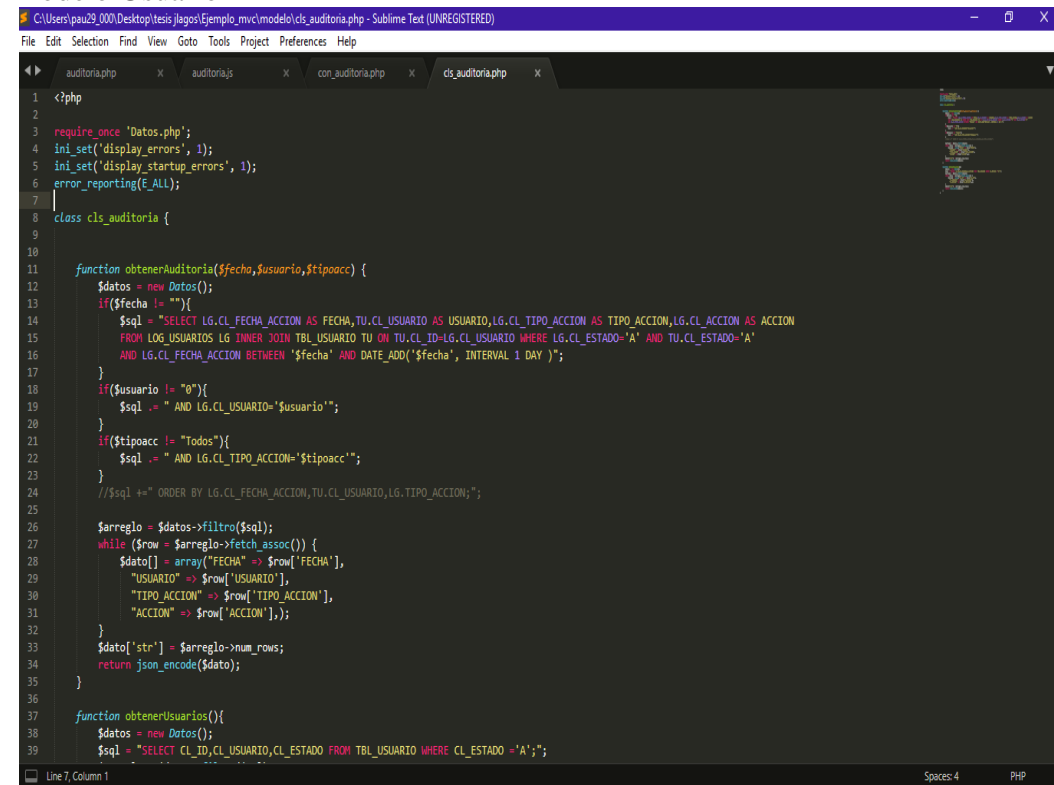
- **JavaScript:** Es un lenguaje de desarrollo orientado a objetos, debido a esto requiere un intérprete que entienda este tipo de lenguaje con el fin de ser comprendido por el procesador. JavaScript no requiere de compilación ya que es ejecutado en un navegador del lado del cliente, estos navegadores son los encargados de interpretar el código fuente (ver Fig. 22). Es por esta razón que el lenguaje mencionado no se compila se traduce (Hernández Pérez, 2016).



- **PHP:** Es un lenguaje de programación se encuentra dentro de los documentos HTML y se ejecuta del lado del servidor, es un lenguaje de código abierto que se le pueden añadir mejoras y compartirlas con los demás usuarios (Hernández Pérez, 2016).

Puede funcionar en casi todos los sistemas operativos, por lo que es un lenguaje de multiplataforma, esto, permite crear páginas dinámicas (ver Fig. 23).

Modelo Usuario PHP



```
1 <?php
2
3 require_once 'Datos.php';
4 ini_set('display_errors', 1);
5 ini_set('display_startup_errors', 1);
6 error_reporting(E_ALL);
7
8 class cls_auditoria {
9
10
11     function obtenerAuditoria($fecha,$usuario,$tipoacc) {
12         $datos = new Datos();
13         if($fecha != ""){
14             $sql = "SELECT LG.CL_FECHA ACCION AS FECHA,TU.CL_USUARIO AS USUARIO,LG.CL_TIPO_ACCION AS TIPO_ACCION,LG.CL_ACCION AS ACCION
15 FROM LOG_USUARIOS LG INNER JOIN TBL_USUARIO TU ON TU.CL_ID=LG.CL_USUARIO WHERE LG.CL_ESTADO='A' AND TU.CL_ESTADO='A'
16 AND LG.CL_FECHA_ACCION BETWEEN '$fecha' AND DATE_ADD('$fecha', INTERVAL 1 DAY)";
17         }
18         if($usuario != "0"){
19             $sql .= " AND LG.CL_USUARIO='$usuario'";
20         }
21         if($tipoacc != "Todos"){
22             $sql .= " AND LG.CL_TIPO_ACCION='$tipoacc'";
23         }
24         // $sql += " ORDER BY LG.CL_FECHA_ACCION,TU.CL_USUARIO,LG.TIPO_ACCION";
25
26         $arreglo = $datos->filtro($sql);
27         while ($row = $arreglo->fetch_assoc()) {
28             $dato[] = array("FECHA" => $row['FECHA'],
29                 "USUARIO" => $row['USUARIO'],
30                 "TIPO ACCION" => $row['TIPO_ACCION'],
31                 "ACCION" => $row['ACCION'],);
32         }
33         $dato['str'] = $arreglo->num_rows;
34         return json_encode($dato);
35     }
36
37     function obtenerUsuarios(){
38         $datos = new Datos();
39         $sql = "SELECT CL_ID,CL_USUARIO,CL_ESTADO FROM TBL_USUARIO WHERE CL_ESTADO = 'A'";
```

Figura 23. Interfaz PHP

Elaborado por: Diego Gualpa, José Lagos.

- **IDE Arduino:** El lenguaje de programación de Arduino está basado en un lenguaje C y C++, su entorno de desarrollo es aplicado en las placas de Arduino (ver Fig. 24). Para compilar el programa en este software, se deberá primero verificar si no existe errores y luego se lo cargará al dispositivo, esto se lo realiza utilizando ciertos botones que incluye el IDE Arduino (Cahuna & Pari, 2017).

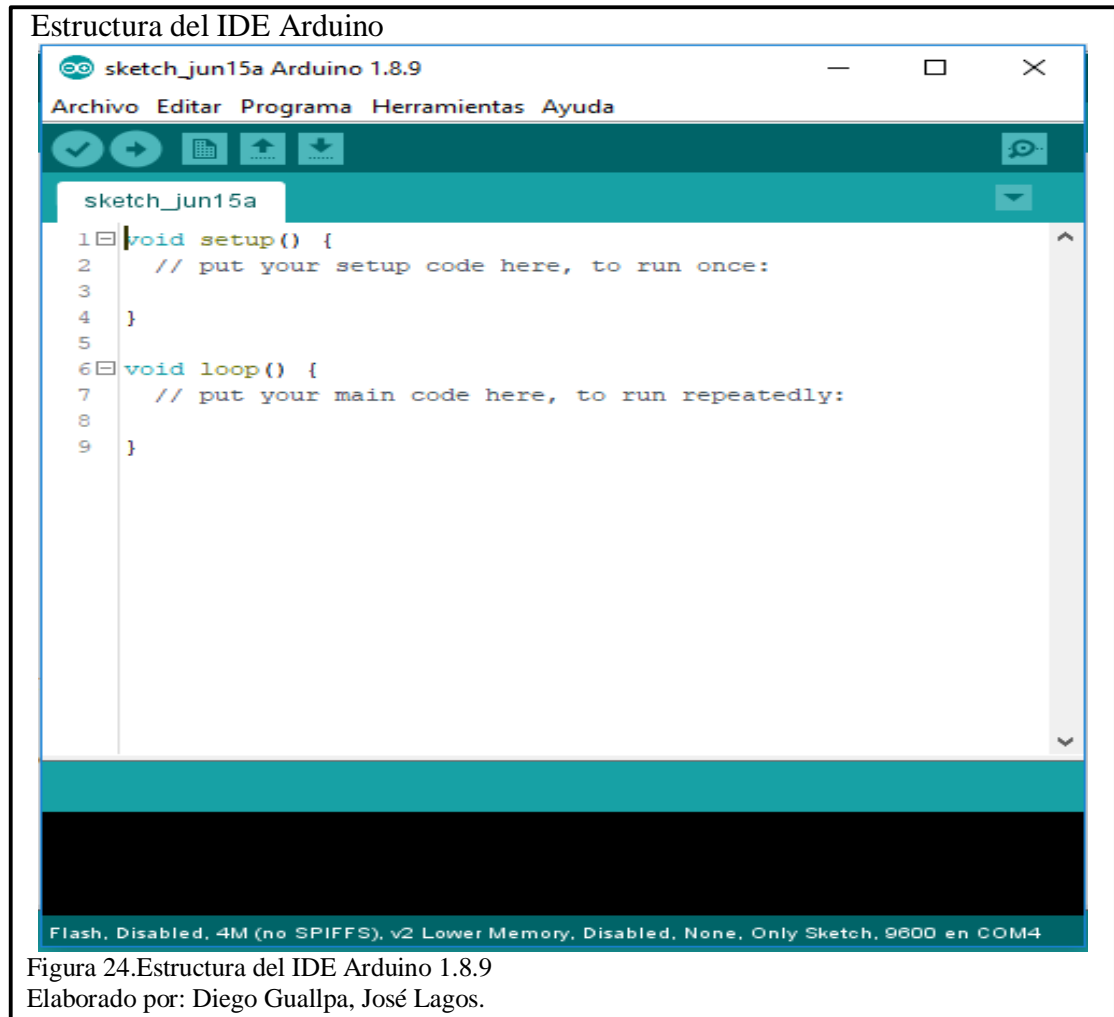


Figura 24.Estructura del IDE Arduino 1.8.9
Elaborado por: Diego Gualpa, José Lagos.

El programa de IDE Arduino se compone de tres secciones a continuación se indicarán cada una de ellas:

- **Sección de declaraciones:** En esta sección se declararán las variables globales del programa y contendrán librerías y ficheros que se utilizarán en el desarrollo del proyecto.
- **Sección void setup:** Se lo llama una sola vez cuando el programa empieza, se lo utiliza para inicializar los modos de trabajo de los pines, o el puerto serie. Debe ser incluido en un programa aunque no haya declaración que ejecutar (Ruiz Gutiérrez, 2007).

- **Sección void loop:** Después de llamar a la sección setup (), se ejecutarán las instrucciones contenidas dentro de la función loop () respondiendo de forma continua e infinitas veces hasta que la placa sea desconectada o reseteada (Ruiz Gutiérrez, 2007).

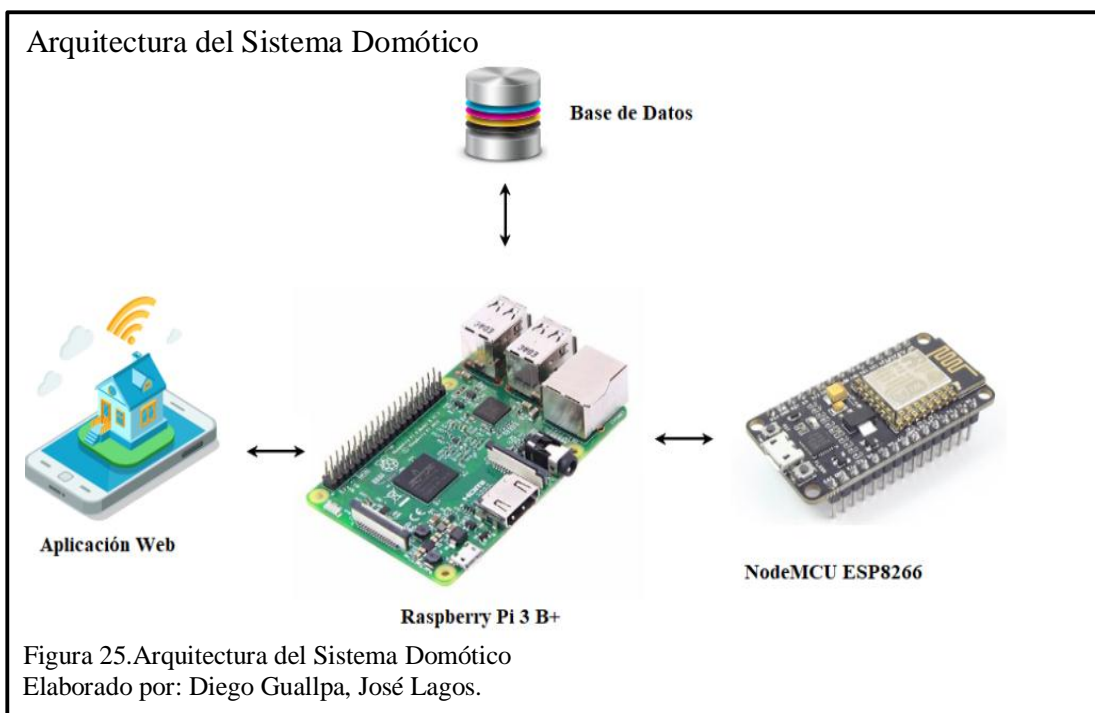
CAPITULO 2

ANALISIS DE HADWARE Y SOFTWARE

Se realizará el análisis de hardware y software utilizado en el presente proyecto, con esto se sabrá con exactitud si el hardware y software utilizado es el adecuado para la realización del proyecto. A continuación se describe el análisis realizado de cada uno de los componentes antes mencionados.

2.1. Hardware para el Control Domótico

El sistema embebido está conformado por una Raspberry Pi 3 B la justificación de esta decisión de diseño viene dada por la versatilidad que aporta dicho sistema, y su gran flexibilidad (ver Fig. 25). Siendo posible instalar en el mismo una gran variedad de distribuciones Linux de muy variadas características que podrán alojar un servidor. Raspberry Pi 3, aporta una gran potencia de procesamiento que permitirá escalar el proyecto sin necesidad de reestructurar el hardware, y cumplir con los requisitos de disponibilidad y rendimiento (Inés, 2017).



Para definir las especificaciones del proyecto domótico se describirá sus principales componentes: Raspberry PI 3 B, NodeMCU ESP8266, Sensores y actuadores, Gestor de Base de Datos MySQL e Interfaz gráfica.

2.1.1 NodeMCU ESP8266

El NodeMCU ESP8266 es el dispositivo que toma el control de la red de sensores y actuadores. En este dispositivo se ejecutarán códigos que se programaron en el IDE Arduino, este a su vez se encarga de hacer una sincronización con el servidor con el fin de mantener datos reales y así mostrarlos en la interfaz de usuario. Este dispositivo debe estar activo (encendido) y compilando para poder acceder al control de los sensores o actuadores (Ceja José, 2017).

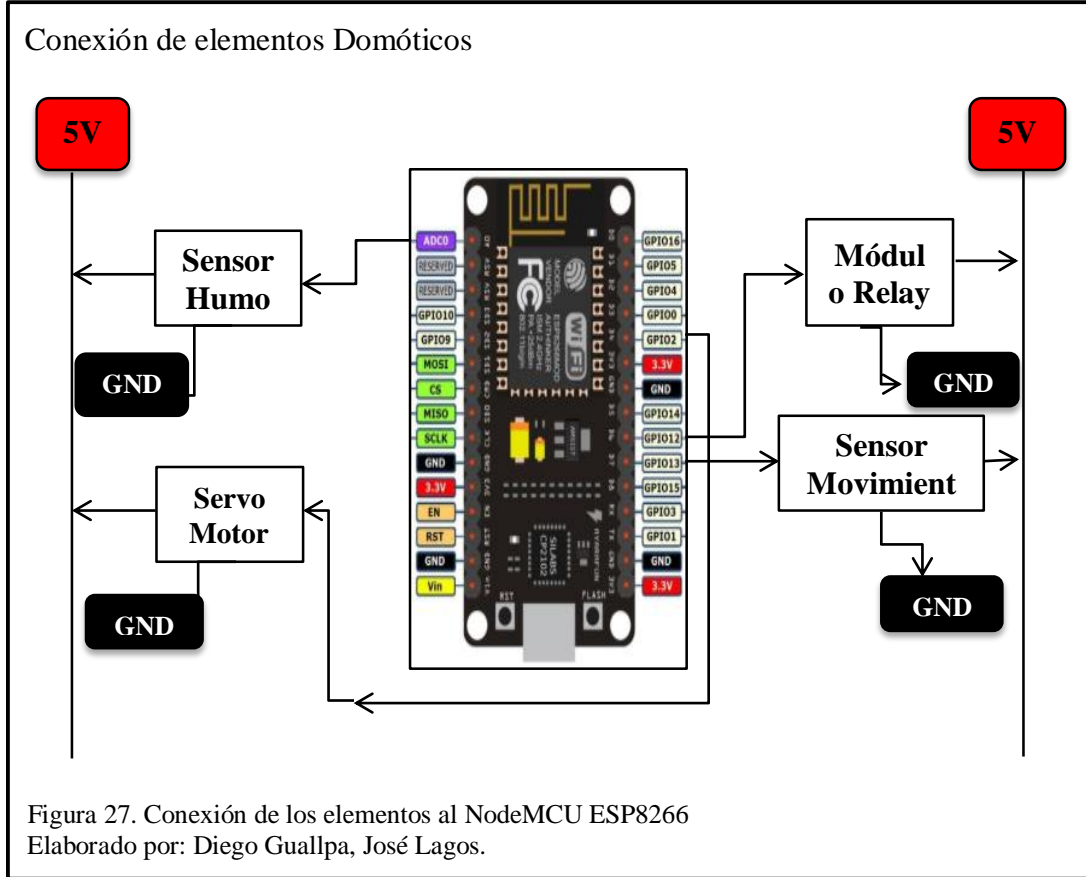
2.1.2 Sensores y Actuadores

Para desarrollar la aplicación domótica se requirió del uso de sensores que permitan conocer su estado o activación y así mediante un controlador tomar una decisión de ejecutar dichos dispositivos (ver Fig. 26). La arquitectura centralizada se caracteriza por llevar el control del elemento del sistema, en este caso el NodeMCU ESP8266, es el encargado de detectar las señales provenientes de los dispositivos sensores o actuadores para así llevar acabo la activación de los dispositivos (Hernández Pérez, 2016).

Características de Sensores y Actuadores		
Tipo Modulo	Tipo de señal	Acción
Sensor Movimiento	Digital	Alerta con mensaje a la aplicación web.
Sensor de Humo	Analógico	Alerta con mensaje ala a la aplicación web.
Actuador Relay	Digital	Encendido y apagado de iluminación

Figura 26. Características de los sensores y actuadores utilizados.
Fuente: (AbcElectrónica, 2015).

Las conexiones de los elementos utilizados en el sistema domótico se muestran en la figura 27.



Los sensores y actuadores se alimentan de 5V, una terminal GND (tierra) y una para la señal de entrada o salida según el NodeMCU lo solicite.

2.1.2.1 Relay

En el presente sistema domótico la petición es enviada por el usuario al NodeMCU, de tal manera que el actuador Relay se encargará de conmutar el estado de un equipo, permitiéndole así realizar la tarea del encendido del foco; La intensidad máxima que puede soportar el Modulo Relay es 10 A (Cabeza, 2018).

2.1.2.2 Sensor de Movimiento PIR

Se ha seleccionado el sensor PIR como dispositivo de detección de movimiento por el motivo que es el más adaptable para la programación con NodeMCU-ESP8266 (Cabeza, 2018). El sensor aplicado en el sistema domótico nos informará de cualquier evento o movimiento detectado, enviando una señal digital al micro controlador, el cual realizará la acción de encendido de iluminación y enviará una notificación al panel de usuario de la página web.

2.1.2.3 Sensor de Humo

Se ha seleccionado el MQ135, este sensor actuará cuando exista gas contaminante mayor al permitido por el sensor. Se emitirá un valor analógico al micro controlador NodeMCU, de tal manera que se enviará una alerta a la interfaz de usuario (página web), con el fin de que se activen las alarmas y así alertar al usuario de posibles incendios, explosiones etc.

2.1.2.4 Servo Motor

El servo motor seleccionado controlara el movimiento de torsión (torque) de la apertura de las persianas del proyecto. En el desarrollo del sistema domotico se controló que sus posiciones angulares fueran de 0° a 180°, 0 grados para cerrar persianas y 180 grados para abrir.

Este servo motor enviará una señal digital al micro controlador NodeMCU, que a su vez será controlado por el usuario mediante la interfaz web, de tal manera que permitirá que el estado sea a modificado a abierto o cerrado.

2.1.3 Gestor de Base de Datos MySQL

Se almacena el estado actual (0,1) de cada uno de los sensores y actuadores, por otra parte se sincroniza entre el NodeMCU y la aplicación. Con esto se guardarán las acciones que se están ejecutando en el sistema domotico (Martinez, 2015).

Se presentan las especificaciones de la base de datos en la Tabla 4.

Tabla 4. Especificación de base de datos

Especificación	Notas
Nombre de la BDD	db_domotik
Sistema de gestión	MySQL
Software de administración	Adminer

Fuente: Diego Gualpa, José Lagos.

2.1.4 Interfaz Gráfica

Se puede decir que la interfaz gráfica es el punto de conexión entre el usuario y el sistema. Con la interfaz generada el usuario puede administrar los diferentes dispositivos que están interactuando con el servidor dentro de la residencia.

En la Tabla 5, se indica el funcionamiento del sistema desarrollado:

Tabla 5. Especificaciones de la interfaz grafica

Especificación	Descripción
Dirección Ip estática	192.168.1.100
Lenguajes de Desarrollo	HTML, CSS, JavaScript, AJAX
Bibliotecas	JQuery 3.1, Bootstrap 4.1
Conexión entre los dispositivos y la interfaz	Un tiempo de conexión de 10 seg. con la base de datos MYSQL
Compatibilidad con los navegadores	Google Chrome, Mozilla Firefox
Acoplamiento para dispositivos móviles	En 3 rangos de pantalla: celular, Tablet, PC

Fuente: Diego Gualpa, José Lagos.

El usuario es capaz de manipular y visualizar los diferentes dispositivos del sistema ya sean actuadores o sensores, dependiendo de su perfil de usuario, ya sea un usuario normal o un administrador. Un punto importante por recalcar es que la interfaz gráfica fue diseñada para poder adaptarse a diferentes tipos de dispositivos inteligentes como computadores, tabletas y celulares, claro está que estos dispositivos deben estar conectados a una red inalámbrica para poder acceder al sistema (Martinez, 2015).

2.2. Software para el Control Domótico

Se trata de un lenguaje interpretado y de sintaxis sencilla que permite un desarrollo rápido y eficiente, el cual se comunicara directamente con el hardware para realizar las distintas tareas (iluminación, movimientos, alarmas). Actuará de manera eficiente ya que su implementación será una API basada en el protocolo HTTP que será consumida por los dispositivos domótico (José Félix, 2012).

2.3. Protocolo de Comunicación

El protocolo de comunicación posibilita que distintos elementos que forman parte de un sistema establezcan comunicaciones entre sí, intercambiando información. En nuestro caso se usará el protocolo de comunicación TCP/IP que permitirá que se enlacen los dispositivos y así puedan acceder a la red. (Porto & Gardey, 2013)

Un protocolo no es otra cosa que el lenguaje en el que se comunican los diversos dispositivos. En el protocolo se define la forma, tamaño y contenido del mensaje para que los dispositivos puedan entenderse. Si el protocolo no está definido, o es distinto en cada dispositivo, no podrán comunicarse (Chicaiza, 2016).

2.3.1. Servidor Web

En el Raspberry Pi B albergar un servidor web ya sea en el área local o en la red no es una tarea muy complicada. En este servidor se puede visualizar páginas web de tal forma que se pueda presentar un resultado hacia el programador con el fin de tener una visión clara del estado de su proyecto. Existen varios servidores de los cuales poseen varios servicios que fueron utilizados en el proyecto los cuales fueron: Apache, MySQL y PHP.

2.3.1.1. Servidor Apache

En el Raspberry Pi 3 este servidor es considerado como una aplicación de servidor web que permite visualizar páginas web. Apache utiliza como protocolo de comunicación el HTTP (puerto 80) el cual permite visualizar páginas dinámicas.

2.3.1.2. MySQL

En su nueva versión se lo llama MariaDB donde se aloja la base de datos de la aplicación domótica. Muy eficiente y reactiva al momento de realizar consultas, inserciones o actualizaciones de datos.

2.3.1.3. PHP

PHP funciona en varios sistemas operativos, de tal manera que se aloja en un servidor remoto o local, este lenguaje proporciona la capacidad de recibir peticiones de un documento HTML y actuar sobre esas peticiones.

CAPITULO 3

DISEÑO E IMPLEMENTACION

Antes de realizar el diseño del prototipo se realizó un estudio básico de como los módulos tenían que ser diseñados y en su programación. Además se realizaron varios diagramas tales como: Casos de uso, Navegación, Arquitectura entre otros. Que sirvieron como guía en corrección de funcionalidades del proyecto.

3.1. Estudio del proyecto

El proyecto a desarrollar presenta varios módulos esenciales: Electrónico, el servidor central y los dispositivos de acceso. En la parte electrónica se toma en cuenta los Sensores, Actuadores y NodeMCU, mientras tanto que el servidor central está constituido por el Raspberry Pi 3b, Apache como servidor web para peticiones y respuestas HTTP. El gestor Mysql está asignado como centralizador de comunicación entre la parte electrónica con los dispositivos de acceso, además de un IDE de desarrollo integrado PHP.

Los dispositivos de acceso se componen de la página web que se puede acceder por medio de computadores de escritorio y portátiles, también se utilizan celulares o tabletas, ya que la página web está realizada con diseño responsivo que se adapta a dispositivos móviles. El sistema web se constituye de varios módulos que son: Inicio de sesión (acceso a usuarios), maneja 2 tipos de perfiles el uno como administrador en el cual realiza todas las tareas de Ingreso, Modificación y eliminación de Usuarios, Estancias, y Objetos (en esta parte se registran los sensores y actuadores a manejar relacionados con el lugar donde se los colocará). Además de un Módulo de auditoria para verificar las diferentes acciones dentro del sistema y el Módulo de control y

manejo de los diferentes Sensores y Actuadores. El perfil de controlador realiza solo las tareas específicas de verificación y control de sensores y actuadores.

Estructura lógica del proyecto.

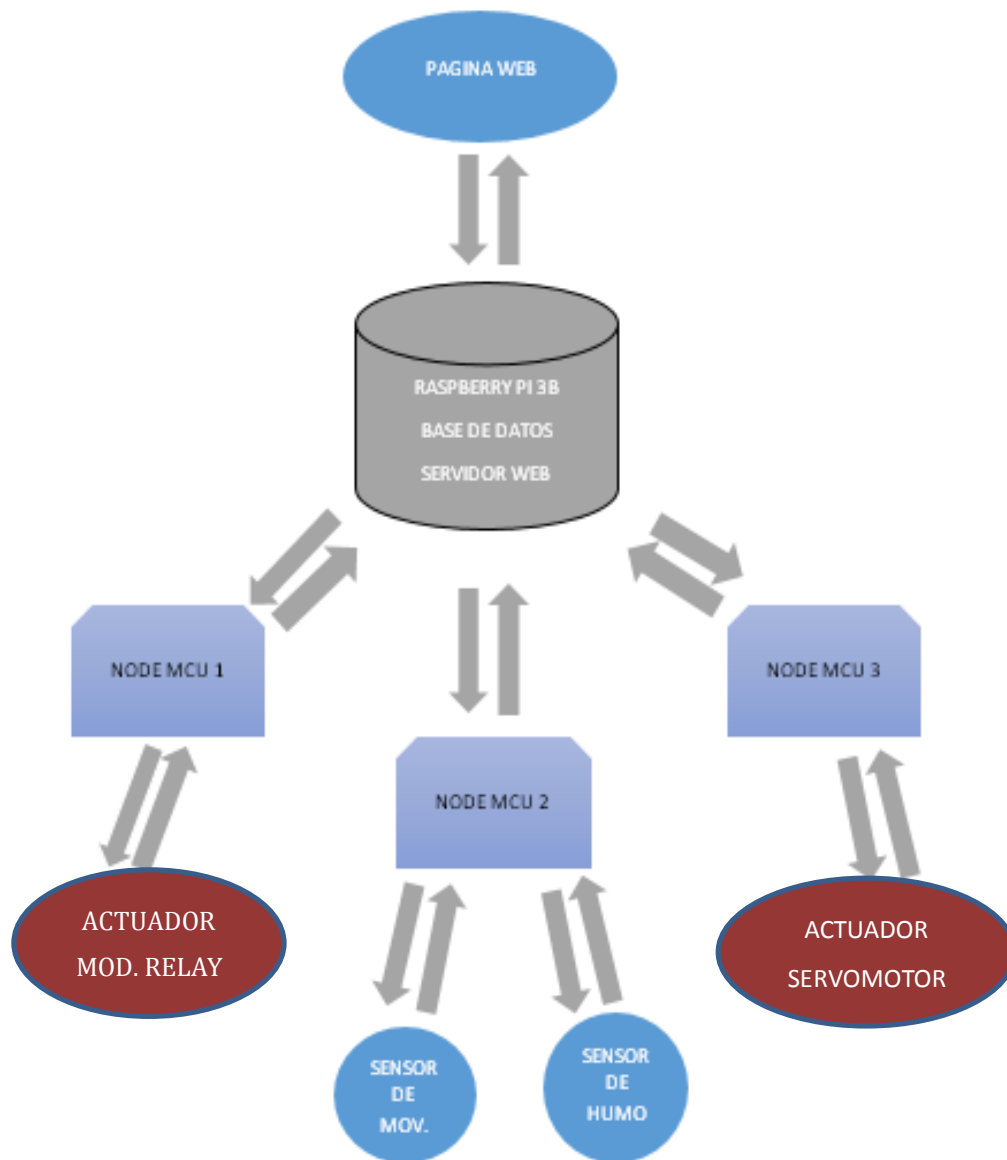


Figura 28. Estructura del proyecto
Elaborado por: Diego Gualpa, José Lagos.

En la Figura 28, se detalla la comunicación del Raspberry con los sensores y actuadores, por medio de NodeMCU. Se utiliza peticiones HTTP GET Y POST desde

el NodeMCU para el intercambio de información entre los dispositivos (sensores, actuadores) con el servidor central.

3.2. Electrónico

El módulo electrónico está compuesto por diferentes componentes en complemento con el microcontrolador integrado llamado NodeMCU, el cual hace la tarea de enviar las peticiones HTTP al servidor Raspberry. Esto lo hace para obtener un estado de activación y con la confirmación del estado retorna una respuesta con el encendido o apagado de los actuadores y una respuesta de valores entregados por los sensores.

El NodeMCU realiza las tareas de control hacia los diferentes dispositivos conectados (sensores y actuadores), además para la programación del mismo se usará el IDE Arduino para mayor facilidad. La misma se realiza conectando mediante el puerto USB directamente. Los módulos utilizados son un Módulo con Relay de 4 canales, un Módulo con Relay de un solo canal, un sensor PIR HCSR-501, un micro servomotor SG-90 que servirá para la apertura de persianas y un sensor de humo MQ-135 para detección de gases uno de ellos puede ser el gas de CO₂ (humo) esto relacionado con la detección de incendios. Para la comunicación del NodeMCU con el servidor Raspberry se inicializan las librerías que permitirán conectarse a la red inalámbrica. En la Figura 29, se observa las librerías que se necesitan para la conexión a la red.

Librerías para conexión a la red Wifi

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266HTTPClient.h>
```

Figura 29. Librerías de conexión por WiFi.
Elaborado por: Diego Gualpa, José Lagos.

Para el envío de datos hacia la página web, NodeMCU se conectará a la red inalámbrica para enviar las peticiones HTTP. En la Figura 30, se muestra los comandos que realizan la función de conexión a la red inalámbrica. Este código se colocará en todos los NodeMCU utilizados en el esquema de hardware.

Código de conexión por Wifi.

```
WiFi.mode(WIFI_OFF); //CUANDO LA CONEXION DEMORA EL WIFI SE APAGA AUTOMATICAMENTE
delay(1000);
WiFi.mode(WIFI_STA); // MODO ESTACION

WiFi.begin(ssid, password);
Serial.println("");

Serial.print("CONECTANDO");
// Wait for connection
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
//SI LA CONEXION FUE EXITOSA IMPRIME LA IP QUE LE ASIGNO EL ROUTER
Serial.println("");
Serial.print("Connected to ");
Serial.println(ssid);
Serial.print("IP address: ");
Serial.println(WiFi.localIP()); //DIRECCION IP ASIGNADA AL NODE MCU
```

Figura 30. Conexión Wifi del NodeMCU.
Elaborado por: Diego Gualpa, José Lagos

3.2.1. Modulo con Relay

El Relay se utiliza en el desarrollo para realizar la función de conmutación de luces, recibe dos valores, 1L o 0L. El paso de corriente por la bobina del Relay se realiza con lógica negativa. En la Figura 31, se visualiza el esquema de conexión del Módulo con Relay y NodeMCU.

Diagrama de conexión del Módulo Relay con NodeMCU.

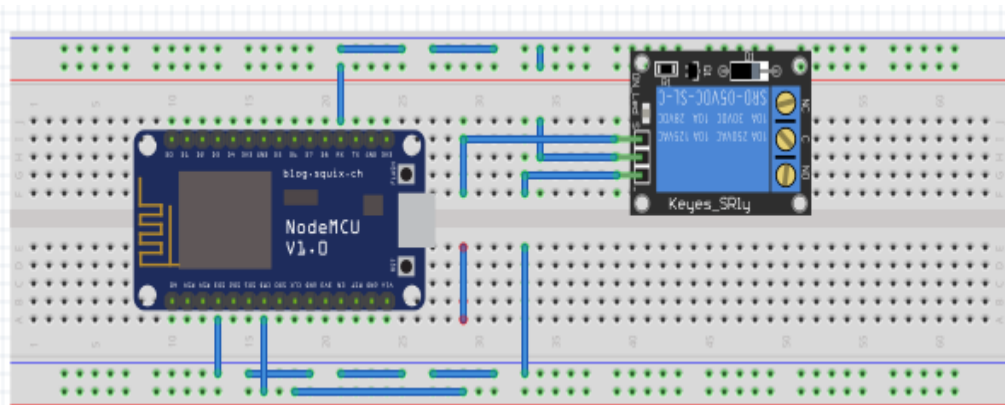


Figura 31. Imagen de Modulo Relay conectado al MCU
Elaborado por: Diego Gualpa, José Lagos.

Programación del NodeMCU para conmutación del módulo con Relay

En la Figura 32, se visualiza la asignación de variables para los pines del NodeMCU donde va a ir conectado el Modulo Relay.

Declaración de variables para los pines de conexión

```
int ledPin = 14; //Lo conectamos a D5, que es el gpio14 enciende y apaga el led del relevador
int ledPin2 = 12; //Lo conectamos a D6, que es el gpio12 enciende y apaga el led del relevador
int ledPin3 = 13; //Lo conectamos a D7, que es el gpio15 enciende y apaga el led del relevador
int ledPin4 = 4; //Lo conectamos a D2, que es el gpio4 enciende y apaga el led del relevador
```

Figura 32. Inicialización de pines para el Modulo Relay
Elaborado por: Diego Gualpa, José Lagos

A continuación en la Figura 33, se indica la un fragmento de programación que hará que el Relay cambie de estado en encendido o apagado. En el fragmento indicado el código que se visualiza es la lógica funcional de todos los Módulos relay.

Programación para activar los Modulo Relay

```
if (payload == "1") {  
    digitalWrite(ledPin, LOW);  
    Serial.println("foco 1 encendido");  
} else {  
    if (payload == "10") {  
        digitalWrite(ledPin, HIGH);  
        Serial.println("foco 1 apagado");  
    }  
}  
  
if (payload == "2") {  
    digitalWrite(ledPin2, LOW);  
    Serial.println("foco 2 encendido");  
} else {  
    if (payload == "9") {  
        digitalWrite(ledPin2, HIGH);  
        Serial.println("foco 2 apagado");  
    }  
}  
}
```

Figura 33. Programación para activar los Modulo Relay.
Elaborado por: Diego Gualpa, José Lagos.

Finalmente, en la Figura 34 se muestra la parte final de la programación donde se realiza la petición http al servidor y la respuesta que devuelve el mismo hacia el NodeMCU mediante la variable *payload*.

Petición hacia el servidor

```
HTTPClient http;  
String Link;  
  
Link = "http://192.168.1.100/luces.txt";  
  
http.begin(Link);    //ESPECIFICA EL DESTINO DE LA PETICION HTTP  
  
int httpCode = http.GET();    //ENVIA LA PETICION  
String payload = http.getString();    //OBTIENE LA RESPUESTA DEL SERVIDOR  
  
Serial.println(httpCode);    //IMPRIME EL CODIGO OBTENIDO DE RESPUESTA A LA PETICION  
Serial.println(payload);    //IMPRIME EL RESULTADO DE LA PETICION LA CADENA DE RESULTADO  
}
```

Figura 34. Código de la petición hacia el servidor web y su respuesta.
Elaborado por: Diego Gualpa, José Lagos.

Tomar en cuenta que para verificar el funcionamiento del código que se programa en el NodeMCU, se debe abrir la consola de verificación del IDE. Para esto se debe inicializar un valor de baudios de 115200 este valor permite comunicar el NodeMCU con el servidor donde se encuentra alojado el sistema.

En la Figura 35, se puede observar el valor de comunicación serial establecido cuando se inicie la consola de pruebas.

Valor de Baudios NodeMCU

```
delay(1000);  
Serial.begin(115200);  
WiFi.mode(WIFI_OFF);  
delay(1000);  
WiFi.mode(WIFI_STA);
```

Figura 35. Valor de baudios para la comunicación de NodeMCU.
Elaborado por: Diego Gualpa, José Lagos.

Vale la pena recalcar, que la configuración de valor de baudios se realiza para todos los sensores y actuadores utilizados en este proyecto.

3.2.2. Sensor de Presencia PIR HCSR-501

Se utiliza en el desarrollo para realizar la función de detección de movimiento, recibe dos valores 1L o un 0L, el sensor PIR trabaja con lógica negativa. Para la detección de movimiento. Además, se añade otro Modulo Relay de un solo canal para verificar el estado que cambia cuando se detecta una presencia.

En la Figura 36, se observa la conexión del sensor PIR con el NodeMCU para su funcionamiento correcto.

Diagrama de conexión del sensor PIR, El Modulo Relay y el NodeMCU.

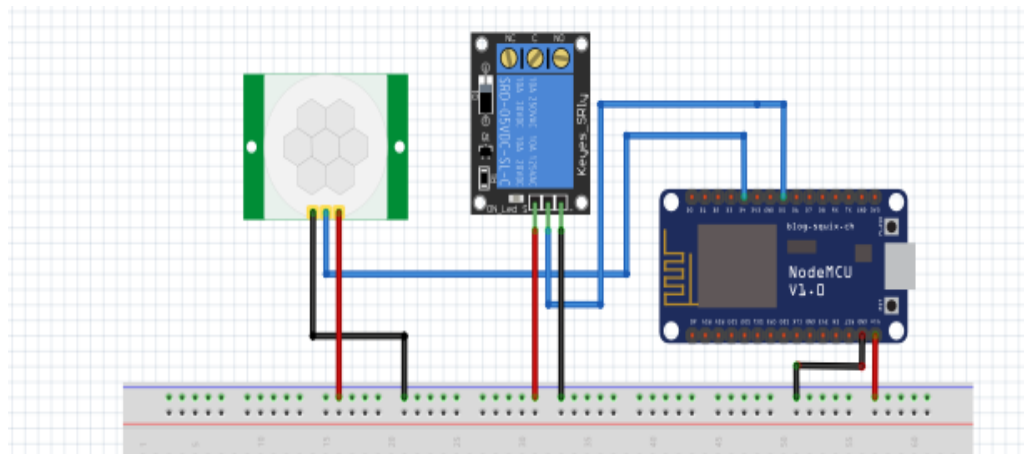


Figura 36. Esquema de conexión del MCU con el sensor PIR y el módulo Relay
Elaborado por: Diego Gualpa, José Lagos.

Programación del NodeMCU para funcionamiento del sensor PIR.

En el código para iniciar el detector de presencia PIR se indica a que pin se conecta y el valor de estado que el micro controlador lo lee. Como primera tarea será inicializar las variables de los pines donde será conectado el Modulo Relay y sensor PIR.

En la Figura 37, se visualiza la asignación de variables a los pines del MCU donde van a ir conectados el Modulo Relay de un canal y el sensor PIR.

Asignación de variables para los pines de conexión.

```
int rele = 12; // Digital pin D6
int sensor = 13; // Digital pin D7
int state=0; //variable que guarda el cambio de estado del PIR
int estado = 0;
```

Figura 37. Asignación de valores de pines para la conexión del sensor PIR y el módulo Relay.
Elaborado por: José Lagos y Diego Gualpa.

En la Figura 38, se muestra la programación del sensor de movimiento para la detección de movimiento. En el caso que exista movimiento el Modulo Relay de un canal se

activará y la bombilla se encenderá por un periodo de tiempo. Una vez finalizado el tiempo de detección de presencia, el módulo Relay se apagará automáticamente.

Código del sensor PIR

```
state = digitalRead(sensor);
Serial.println("State");
Serial.println(state);
delay(500);
if (state == 1) {
    digitalWrite (rele, LOW);
    Serial.println("MOVIMIENTO DETECTADO!!!");
    //state = 0;
    estado = 1;
    delay(1000);
}
else {
    digitalWrite (rele, HIGH);
    Serial.println("NO SE DETECTA MOVIMIENTO!!!");
    estado = 0;
    delay(1000);
}
```

Figura 38. Programación del sensor PIR.
Elaborado por: Diego Gualpa, José Lagos.

En la Figura 39, se muestra la parte final de la programación donde se realiza la petición http al servidor y la respuesta que devuelve el mismo hacia el NodeMCU. La diferencia en el anterior sensor es que realiza una petición GET en la cual envía la variable donde almacena el estado del sensor cuando detectó una presencia. La misma que enviará hacia el servidor web para realizar tareas posteriores de funcionamiento con la página web. De igual forma utiliza una variable *payload*.

Código de petición GET

```
String ADCData,getData, Link;
ADCData=String(estado);

//REALIZA PETICION GET
getData = "?valor=" +ADCData; //ENVIA LA VARIABLE CON EL VALOR DEL ESTADO DEL SENSOR
Link = "http://192.168.1.100/enviodelesp.php" + getData;

http.begin(Link);    //ESPECIFICA EL DESTINO DE LA PETICION QUE SE VA A REALIZAR

int httpCode = http.GET();    //ENVIA LA PETICIÓN
String payload = http.getString();    //OBTIENE EL PAYLOAD DE RESPUESTA

Serial.println(httpCode);    //IMPRIME EL CODIGO RETORNADO
Serial.println(payload);    //IMPRIME LA RESPUESTA DEL SERVIDOR

http.end();    //FINALIZA CONEXION CON EL SERVIDOR

delay(10000);    //OBTIENE LOS DATOS CADA 10 SEGUNDOS.
```

Figura 39. Código de petición GET hacia el servidor web.
Elaborado por: Diego Guallpa, José Lagos.

3.2.3. Micro Servo Motor SG-90

Se utiliza en el desarrollo para realizar la función de apertura de persianas recibe dos valores de rotación de ángulos que van desde 0° hasta 90°, que hacen que el motor abra o cierre las cortinas. El comportamiento del servomotor es en forma mecánica lo que hace que su funcionamiento sea vulnerable hacia cualquier sobrecarga de voltaje o ante cualquier manipulación manual.

En la Figura 40, se observa la conexión del Servomotor con el NodeMCU para su respectivo funcionamiento.

Conexión del servomotor

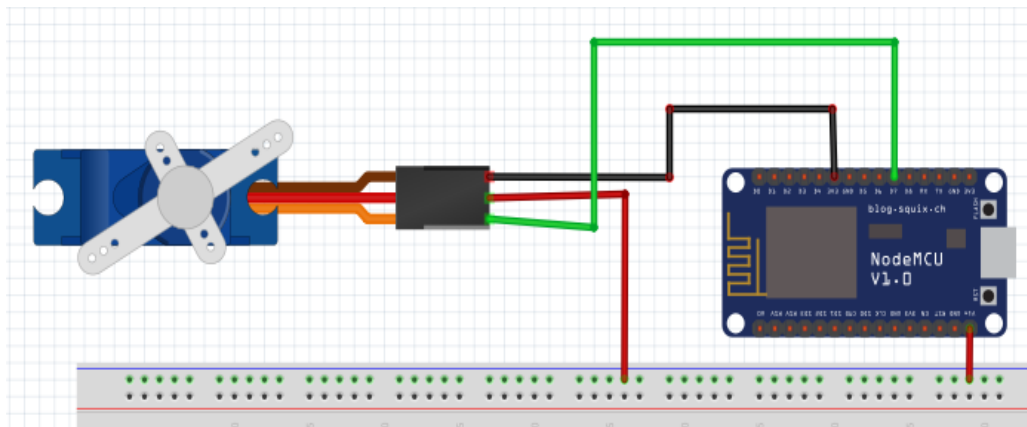


Figura 40. Conexión de NodeMCU con el servomotor
Elaborado por: Diego Gualpa, José Lagos.

Programación del NodeMCU con el servomotor SG-90.

En el código para activar o desactivar el servomotor SG-90 se indica a que pin se conecta el y el valor de rotación del Angulo en este caso será cero grados que es el estado de cerrado. El estado de abierto cambia según el dato de respuesta que llegue desde el servidor web.

En la Figura 41, se visualiza la declaración y asignación de variables de los pines del MCU donde va a ir conectado el servomotor.

Pin de conexión

```
servo.attach(2); //CONECTAMOS AL D4 PIN 2  
  
servo.write(0); //INICIA EL VALOR EN CEROGRADES.
```

Figura 41. Código que asigna el pin de conexión para el servomotor.
Elaborado por: Diego Gualpa, José Lagos

En la Figura 42, se muestra la programación que hará que se active el servomotor dependiendo si la respuesta del servidor es uno el motor girará 90° caso contrario regresa a su posición inicial es decir 0°.

Rotación del servomotor

```
if(payload == "1"){  
  servo.write(360);  
  Serial.println("PERSIANA ABIERTA");  
  delay(1000);  
}else{  
  servo.write(0);  
  Serial.println("PERSIANA CERRADA");  
  delay(1000);  
}
```

Figura 42. Código que realiza la rotación del motor.
Elaborado por: Diego Gualpa, José Lagos.

En la Figura 43, se muestra la parte final de la programación donde se realiza la petición http al servidor y la respuesta que devuelve el mismo hacia el NodeMCU para activar y desactivar el servomotor, la respuesta la recibe por medio de la variable *payload*.

Petición HTTP

```
HTTPClient http;
String Link;

Link = "http://192.168.100.100/valor.txt";

http.begin(Link);    //ESPECIFICA EL DESTINO DE PETICIÓN.

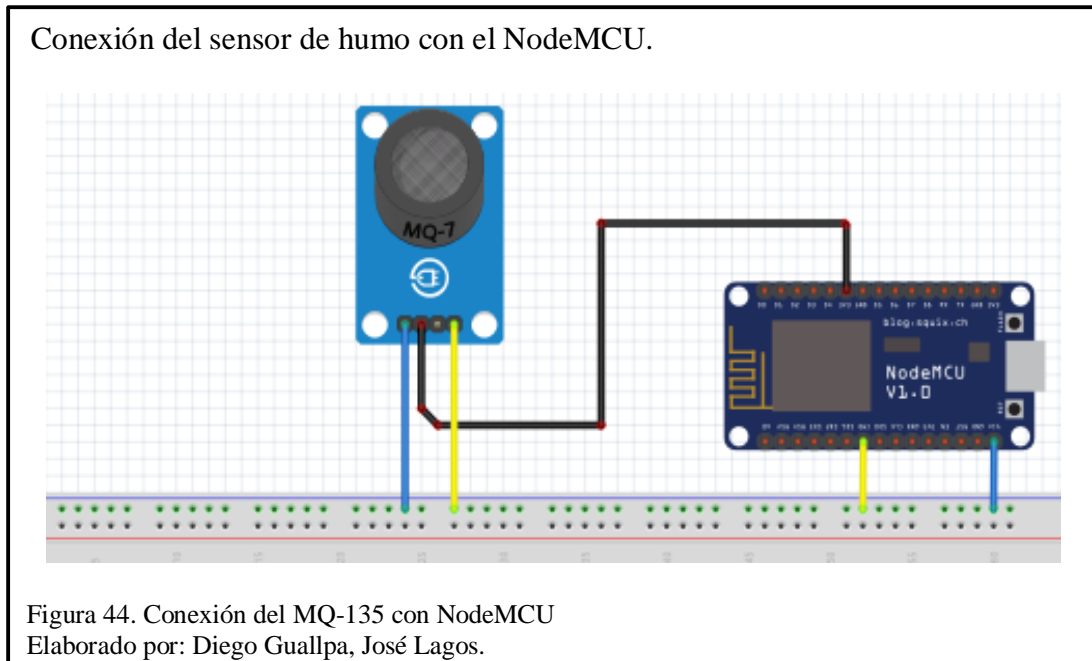
int httpCode = http.GET();    //ENVIA LA PETICIÓN.
String payload = http.getString();    //OBTIENE LA RESPUESTA DEL SERVIDOR EN EL PAYLOAD.

Serial.println(httpCode);    //IMPRIME EL CODIGO DE RETORNO.
Serial.println(payload);    //IMPRIME EL PAYLOAD.
if(payload == "1"){
    servo.write(90);
    delay(1000);
}else{
    servo.write(0);
    delay(1000);
}
http.end();    //SE FINALIZA LA CONEXIÓN.
delay(1000);
```

Figura 43. Código que envía la petición http al servidor.
Elaborado por: Diego Gualpa, José Lagos.

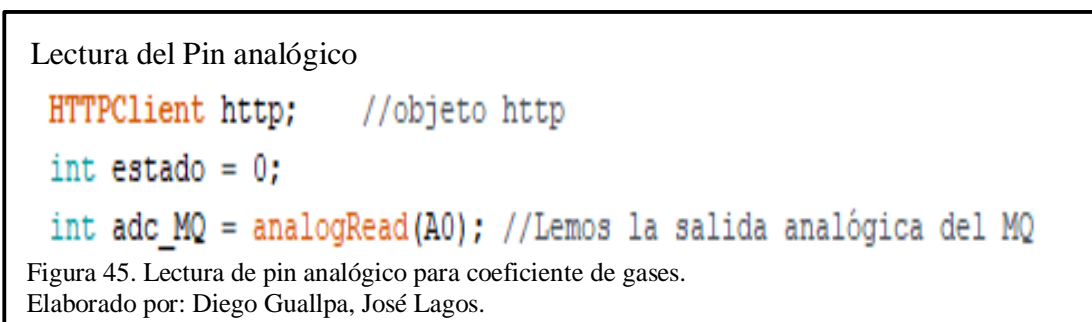
3.2.4. Sensor de humo MQ-135

El sensor de humo MQ-135 se utiliza en el desarrollo para realizar la función de detección de gases en el ambiente la razón principal para usar este tipo de sensor es para la detección de humo. El funcionamiento del sensor se basa en dos formas Analógica y Digital. En el proyecto se usa el pin de entrada analógico, en el cual el sensor despliega un valor inicial para determinar que no existen gases dentro del ambiente. La activación del sensor sucede cuando el valor emitido es mayor al valor inicial.



Programación del NodeMCU para controlar el sensor MQ-135

En el código para controlar el sensor MQ-135 se indica a que pin se conecta el y el coeficiente de detección de gases emitido. En la Figura 45, se visualiza la lectura del coeficiente emitido para la detección de humo.



A continuación, en la Figura 46 se puede visualizar el código de programación que sirve para la detección de humo. La misma se realiza según el valor de coeficiente que el sensor detecte si el valor detectado es mayor al que emite por defecto emitirá un mensaje de alerta con el resultado del valor inicial.

Código de funcionamiento del sensor

```
HTTIClient http;    //objeto http
int estado = 0;
int adc_MQ = analogRead(A0); //Lemos la salida analógica del MQ
float voltaje = adc_MQ * (5.0 / 1023.0); //Convertimos la lectura en un valor de voltaje

Serial.print("adc:");
Serial.print(adc_MQ);
Serial.print("    voltaje:");
Serial.println(voltaje);
if (voltaje > 1) {
    Serial.println("humo detectado");
    estado = 1;
}
else {
    Serial.println("no existe humo!!!");
    estado = 0;
}

String ADCData,getData, Link;
ADCData=String(estado);
```

Figura 46. Código de funcionamiento del detector de gases.
Elaborado por: Diego Gualpa, José Lagos.

En la Figura 47, muestra la parte final de la programación donde se realiza la petición http al servidor para enviar el valor de coeficiente de gas. Y a su vez la respuesta que devuelve el mismo hacia el NodeMCU para poder activar y desactivar el servomotor, la respuesta la recibe por medio de la variable *payload*.

Código de envío Http

```
//ENVIA VALOR DE ESTADO HACIA EL SERVIDOR.
getData = "?humo=" +ADCData;
Link = "http://192.168.1.100/enviohumo.php" + getData;

http.begin(Link);    //Specify request destination

int httpCode = http.GET();    //Send the request
String payload = http.getString();    //Get the response payload

Serial.println(httpCode);    //Print HTTP return code
Serial.println(payload);    //Print request response payload

http.end(); //Close connection
delay(5000);
```

Figura 47. Código de envío Http.
Elaborado por: Diego Gualpa, José Lagos.

3.3. Servidor central

El servidor central realiza las tareas de conexión entre los dispositivos y la aplicación web, el servidor central es el RPI 3B. Dependiendo las necesidades de operación del Raspberry se lo puede configurar como un servidor doméstico o un dispositivo orientado a procesos de empresas con objetivos a largo plazo.

Los servicios del servidor central deben ser inicializados en forma secuencial, se debe instalar varios programas como los que se describen a continuación:

1. Se debe descargar el sistema operativo adecuado en su última versión, en este caso será Raspbian.

2. Una vez descargado el S.O se debe guardar en una tarjeta micro SD ultra que tenga mínimo 8 GB de capacidad.
3. Se inserta la tarjeta micro SD y se procede con la instalación del sistema operativo, siguiendo los pasos que muestra en pantalla.
4. Se instala el gestor de base de datos Mysql Server mediante los comandos correspondientes.
5. Se inicia la instalación de PHP, con las sentencias listadas en la Figura 48.

Repositorios de instalación

```
sudo apt-get install php-curl  
sudo apt-get install php-gd  
sudo apt-get install php-imap  
sudo apt-get install php-json  
sudo apt-get install php-mcrypt  
sudo apt-get install php-opcache  
sudo apt-get install php-xmldrpc  
sudo apt-get install php-mysql
```

Figura 48. Comandos de instalación de PHP.
Elaborado por: Diego Gualpa, José Lagos.

1. Se instala apache 2 con la siguiente sentencia `sudo apt-get install apache2`.
2. En la Figura 49, se realiza la instalación repositorios de comunicación de apache2 y Mysql.

Sentencias de instalación de apache2

```
sudo apt-get update  
sudo service apache2 start  
sudo service mysql start
```

Figura 49. Sentencias de inicialización Apache2 y Mysql.
Elaborado por: Diego Gualpa, José Lagos.

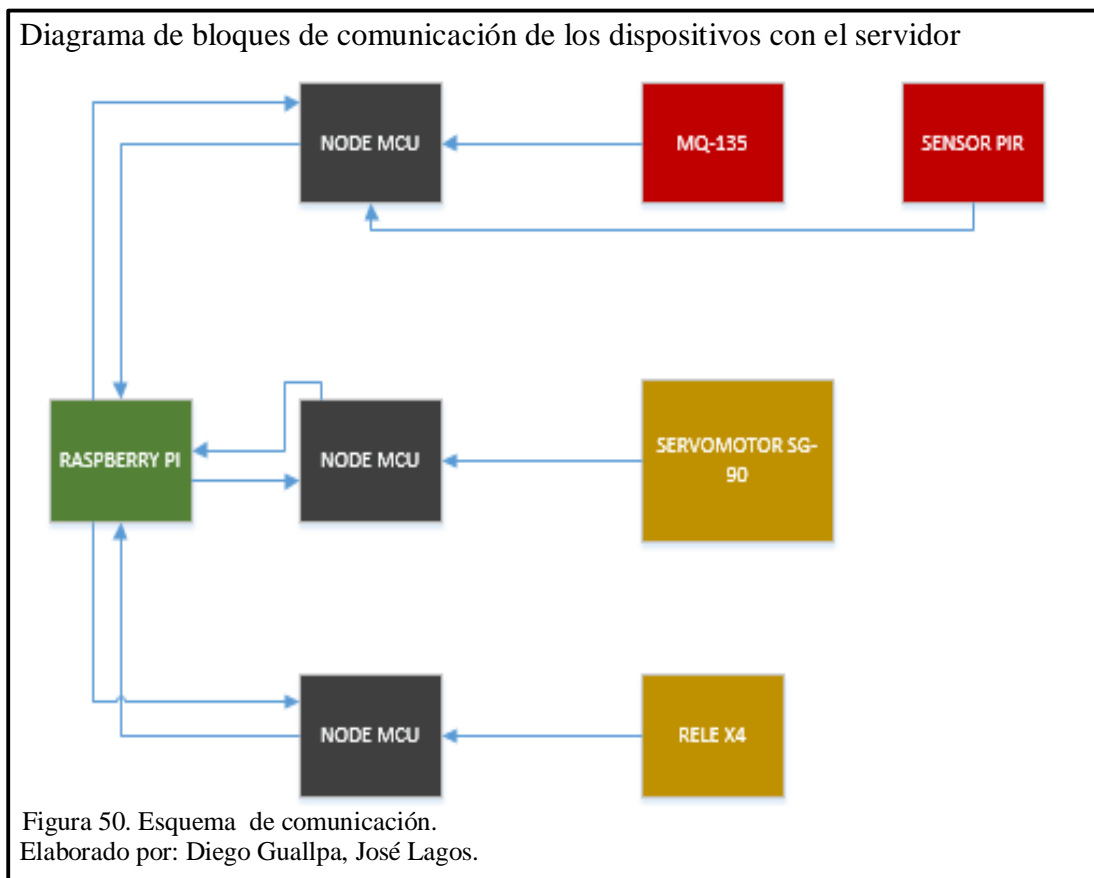
3.4. Software

Es la parte que se encarga de vincular los dispositivos electrónicos con el servidor web, esto lo hace por medio de un conjunto de sentencias, variables, métodos y objetos. El software se encarga de recibir las solicitudes HTTP que envían los NodeMCU y enviar respuesta incluyendo parámetros de activación para los dispositivos electrónicos conectados.

Se usan métodos POST para la respuesta desde el servidor hacia los dispositivos micro controladores y métodos GET. Esto se utiliza para el envío de peticiones desde los MCU hacia el servidor para que el software realice tareas de procesamiento.

3.4.1. Diagrama de Comunicación de Hardware

En la Figura 50, se visualiza el diagrama de comunicación del prototipo, donde se indica la conexión de los dispositivos electrónicos con los NodeMCU. Estos a su vez se vinculan a la Raspberry (servidor) y por un enlace a la red inalámbrica (WIFI), procesa la información, y posteriormente envía respuestas a las peticiones que realizan los NodeMCU.



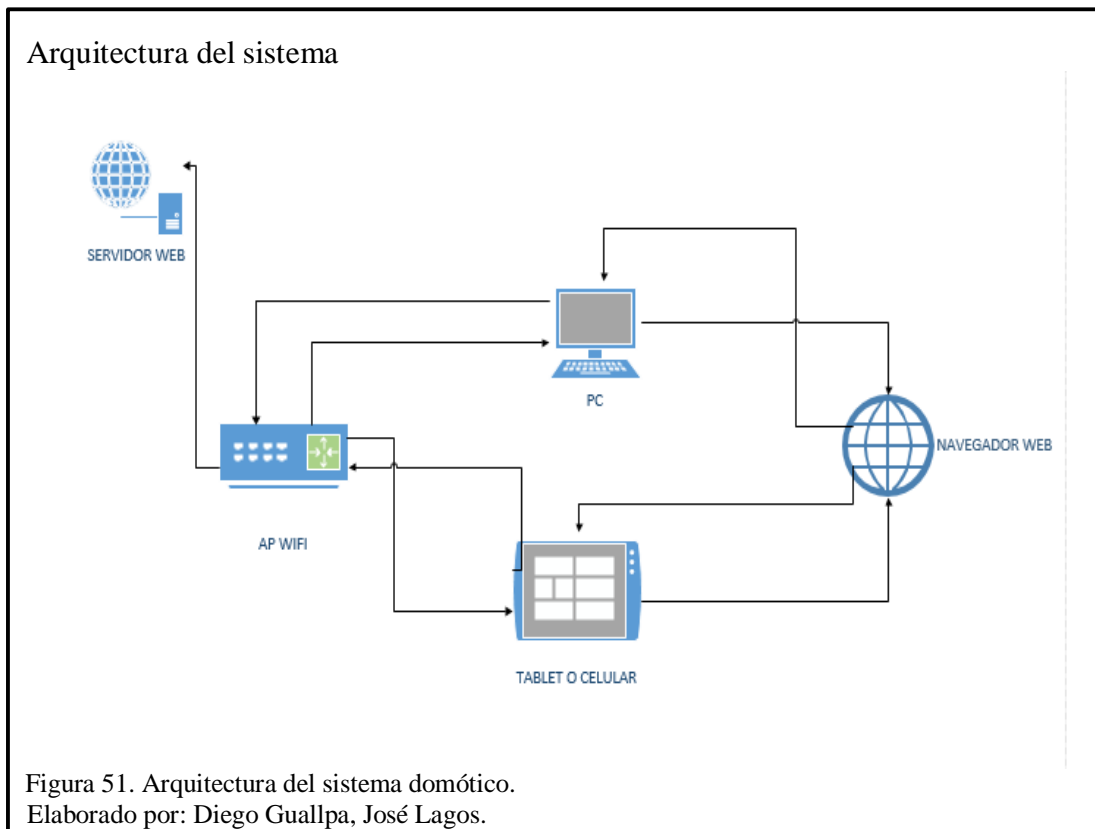
3.4.2. Análisis de Software

Consta de un ambiente web programado en procesador de hipertexto (PHP) y lenguaje de marcas de hipertexto (HTML) además utilización de JavaScript, JQuery y Ajax para procesamiento de funciones hacia la base de datos. Además, el software esta realizado mediante programación por capas (MVC) que separa la vista hacia el usuario con el código backend y las funciones aplicadas hacia el modelo de datos.

3.4.3. Arquitectura del Sistema

En la Figura 51, se visualiza el esquema de la arquitectura del servidor web, el cual por medio de un router inalámbrico establece el enlace para el envío de los datos que serán visualizados desde un navegador en la red local. Además la arquitectura del proyecto se basa en protocolo TCP esto quiere decir que en el envío de peticiones

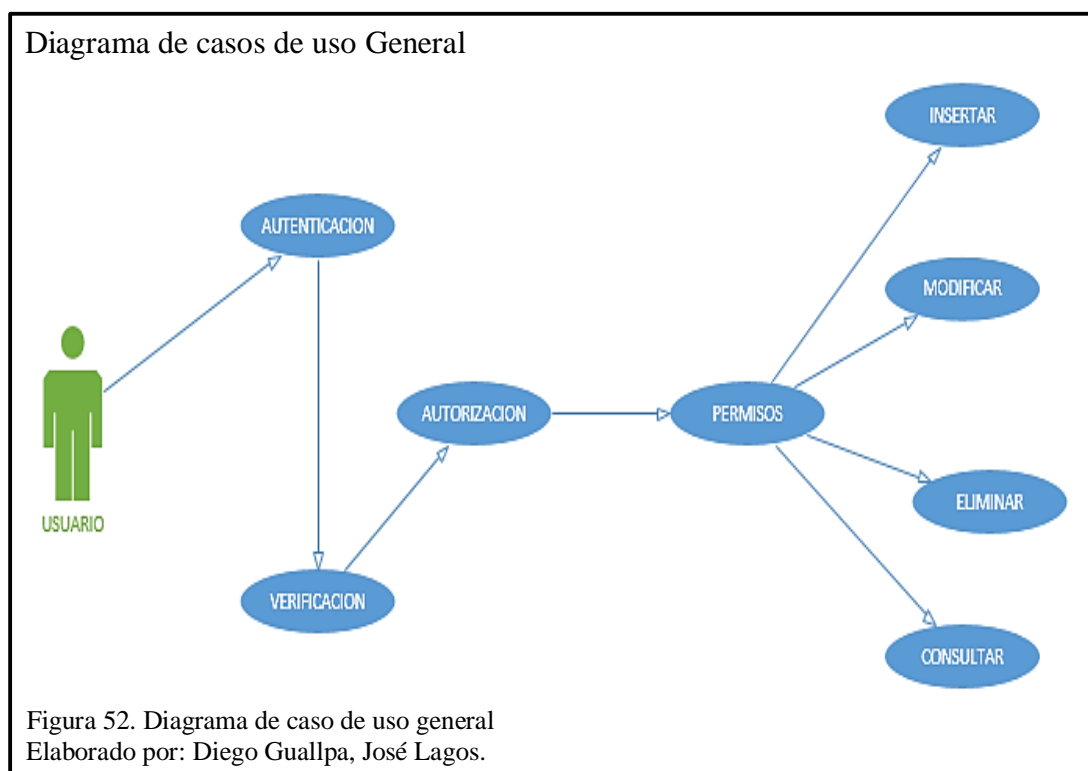
siempre se esperará una respuesta de confirmación del servidor luego de la petición. Esto se conoce comúnmente como acuse de recibo o también llamado ACK. Los ACK'S son las confirmaciones que se envían después de una petición TCP si el destino no envía un ACK de confirmación la petición queda pendiente hasta que esté disponible.



3.4.4. Diagrama de casos de uso

Este tipo de esquema describe el comportamiento de un actor dentro de un sistema. Se utilizan para dar a conocer los diferentes módulos existentes del sistema y ver las reacciones ante eventos propios o de su ambiente. En la Figura 52, se observa el modelo de caso de uso aplicado.

Depende el rol del negocio los diagramas de casos de uso pueden variar en ítems de tareas. También depende de la complejidad de procesos y tareas que se realizarán dentro del sistema que sea implementado en dicho negocio.



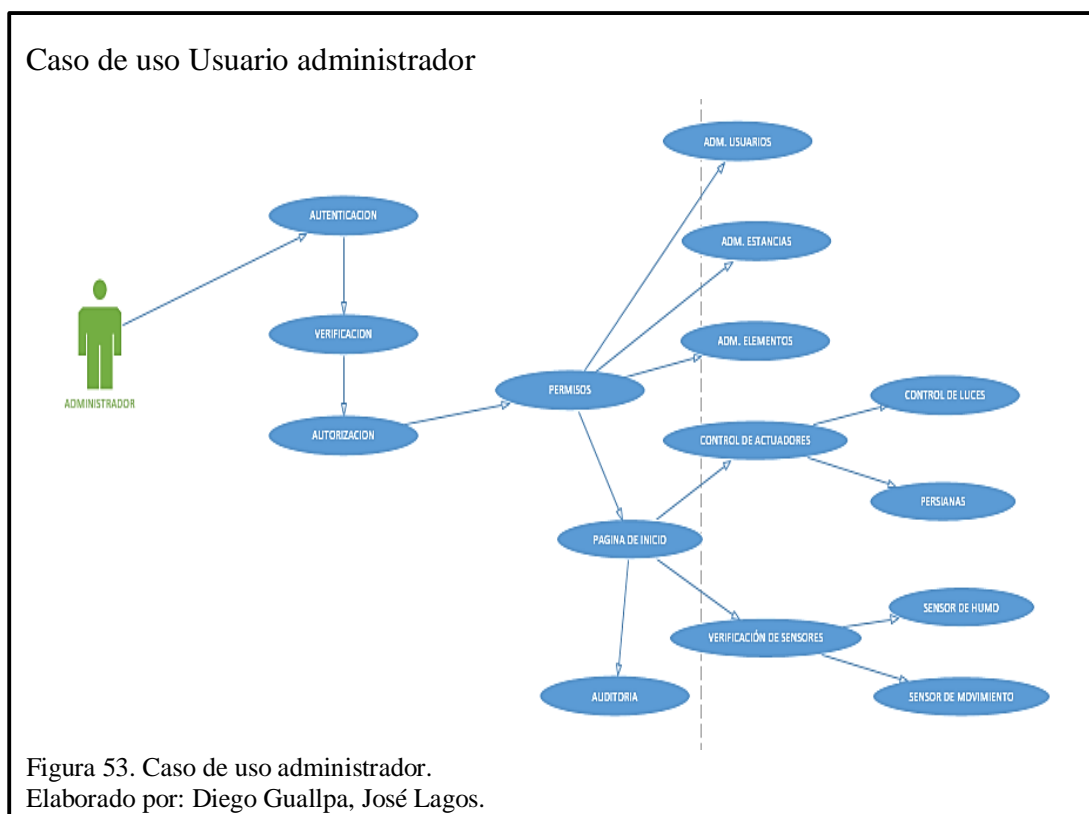
3.4.4.1. Actores o Roles

Los actores o roles son definidos dentro del proyecto dependiendo las características funcionales del mismo. Entre los roles más básicos se encuentran los administradores y los usuarios o invitados los cuales se describen a continuación:

- **Administrador:** Rol que se asigna permisos completos en un sistema, también conocido como súper usuario.
- **Controlador:** Actor con restricciones y permisos de visualización y control.

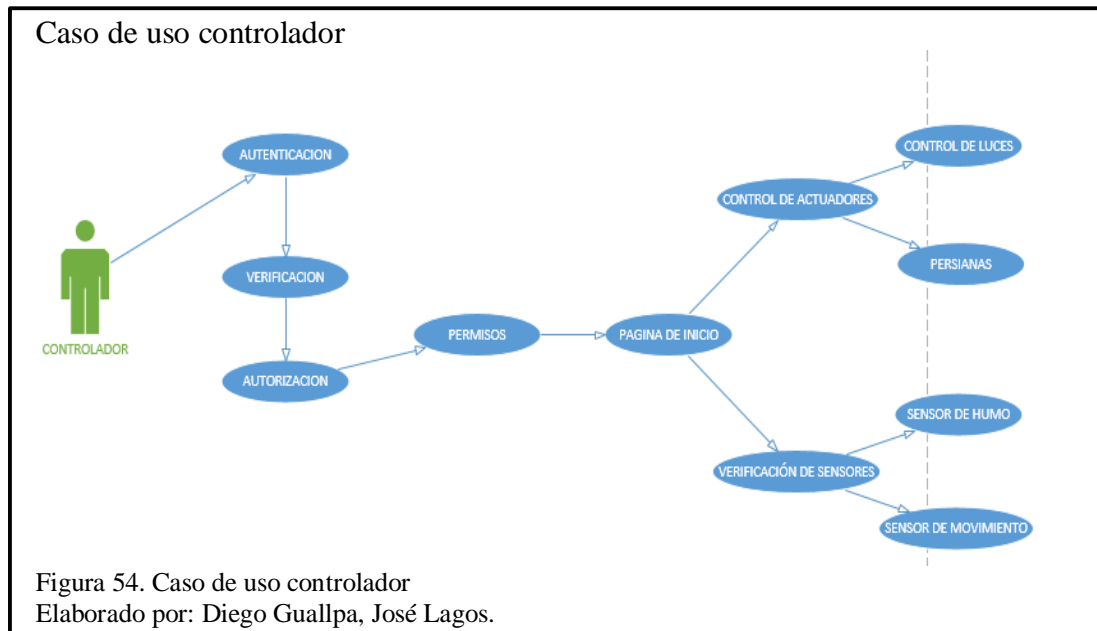
3.4.4.2. Usuario Administrador

En la Figura 53, se puede visualizar como el rol de administrador tiene privilegios frente a otros usuarios como crear, modificar, eliminar y visualizar información.



3.4.4.3. Usuario Controlador

En la Figura 54, se muestra los casos de uso asignados al usuario controlador el cual tiene restricciones con respecto al administrador. Los mismos que son control de dispositivos y monitoreo de los estados respectivos de los mismos.

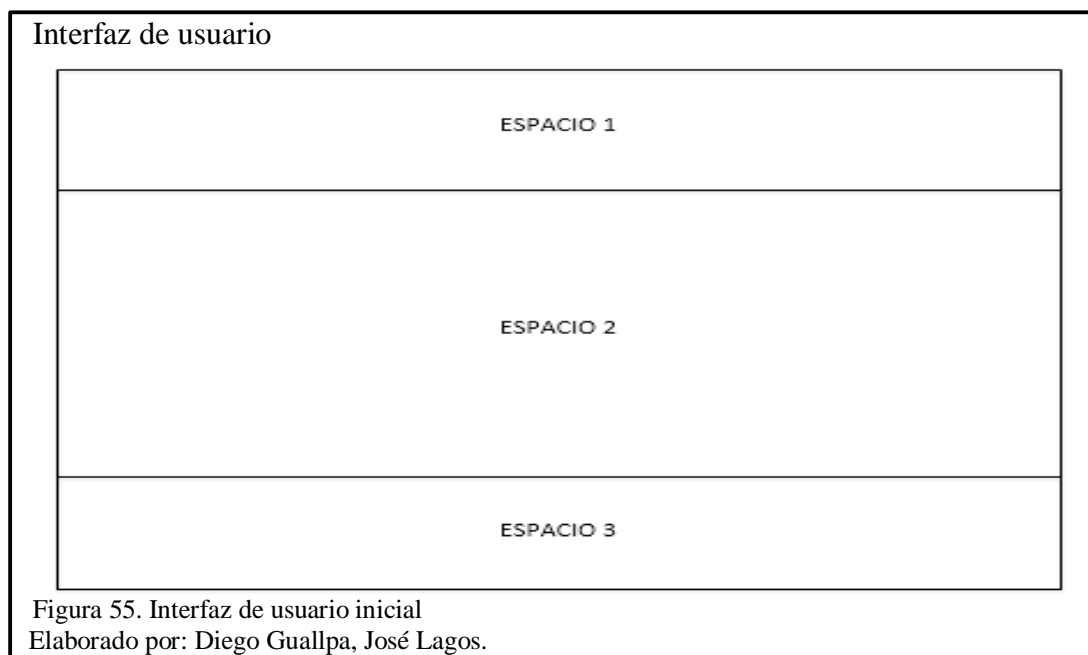


3.4.5. Interfaz de Usuario

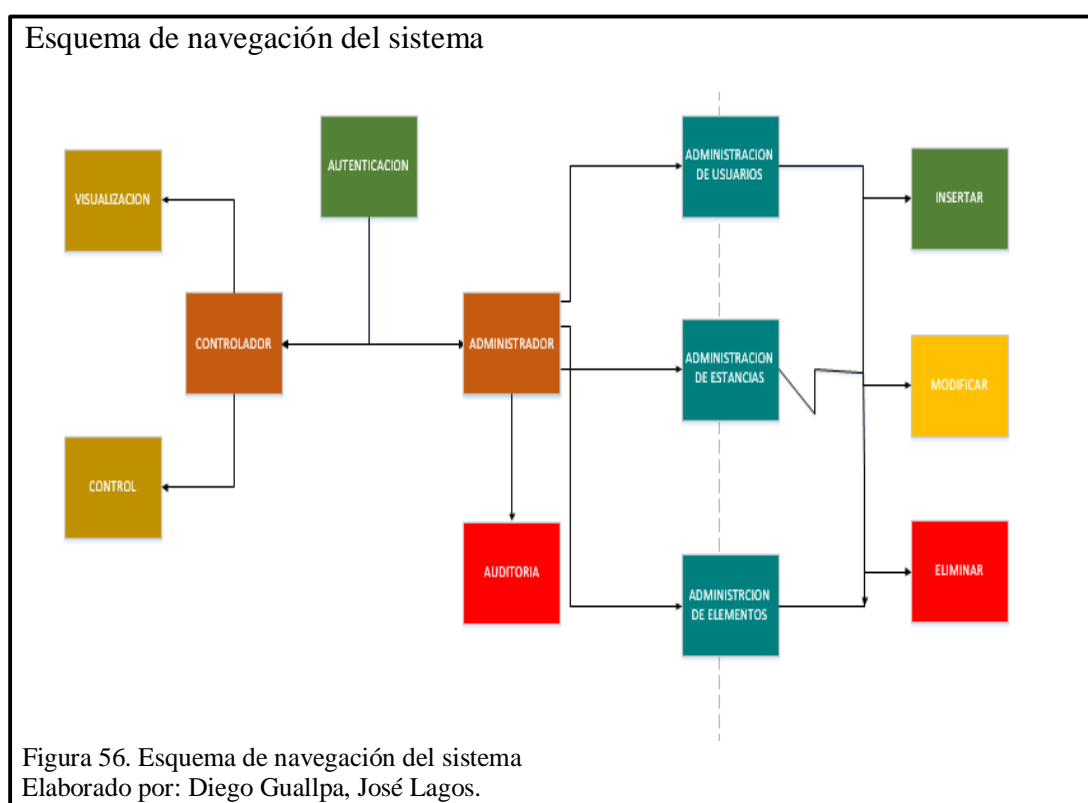
Se define como interfaz de usuario al medio de comunicación entre un computador, maquina o dispositivo y el usuario final, son amigables y de fácil comprensión.

En la Figura 55, se visualiza la división esquemática del entorno web, constituido de la siguiente manera:

- Menú de navegación del sistema en el espacio 1.
- Ubicación de los controles para manipulación de dispositivos en el espacio 2.
- Ubicación de pie de página con el nombre del sistema en el espacio 3.



En la Figura 56, se visualiza el esquema de navegación del sistema, donde se indican los respectivos roles y privilegios antes descritos.



3.4.6. Inicio de Sesión

En la Figura 57, se visualiza la pantalla de acceso a usuarios en el sistema, tiene controles de acceso por medio de un formulario de ingreso de credenciales. Los mismos que son validados por medio de la consulta hacia los registros de la base de datos, verificando si el usuario ingresado es administrador o controlador.

Login del sistema



Usuario

Contraseña

Aceptar

© Domotic-UPS 2018-2019

Figura 57. Pantalla de Login
Elaborado por: Diego Gualpa, José Lagos.

Si el usuario ingresado tiene rol de administrador, se visualiza una opción en la barra de menús superior del sistema, permitiendo la interacción por medio de consultas, modificaciones y eliminaciones de usuarios, elementos y estancias. Además de controlar y verificar estados de sensores y actuadores y acceder al módulo de auditoría.

En la Figura 58, se visualiza la pantalla de inicio la cual contiene los módulos existentes dentro del sistema. Los módulos controlarán cada uno de los dispositivos conectados dentro de la residencia.

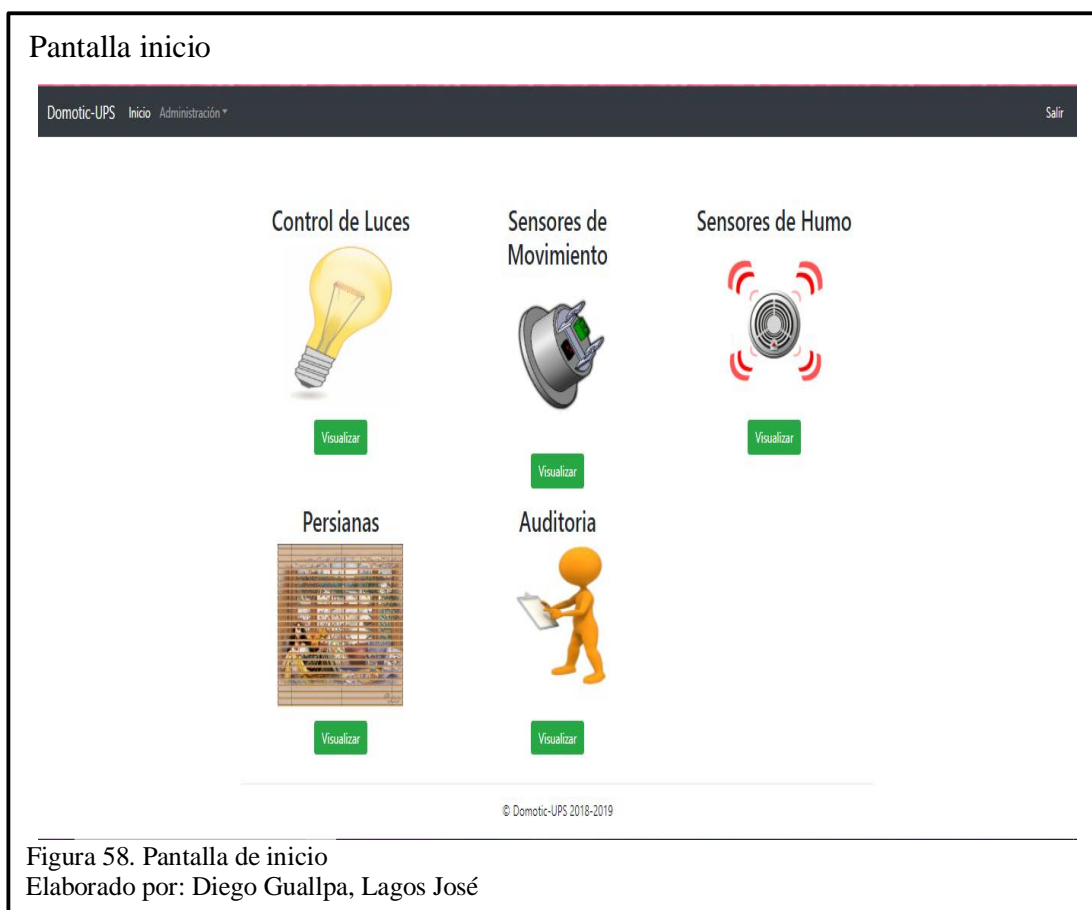


Figura 58. Pantalla de inicio
Elaborado por: Diego Gualpa, Lagos José

3.4.7. CRUD de Usuarios

En la Figura 59, se visualizan los usuarios que fueron ingresados en el sistema, con atributos especiales dependiendo el tipo de permiso que se otorgó (administrador o controlador). Se observa los campos para ingresar los datos correspondientes, y las tareas de modificar y eliminar usuarios.



3.4.8. Pantalla de Estancias

En la Figura 60, se visualizan las estancias registradas en el prototipo, con el estado correspondiente. Además de un módulo de ingresar estancias y un CRUD que permite modificar o eliminar estancia. Esta pantalla en cuestión muestra los lugares existentes dentro de la residencia, y dentro de cada lugar el dispositivo conectado y configurado. Además de que por cada estancia indica el estado actual de la misma es decir si esta activa o inactiva.

Pantalla de visualización de estancias

Lista Estancias

Nombre Lugar	Estado		
Comedor	Activo	Modificar	Eliminar
Dormitorio1	Activo	Modificar	Eliminar
Dormitorio2	Activo	Modificar	Eliminar
Dormitorio3	Activo	Modificar	Eliminar
Dormitorio4	Activo	Modificar	Eliminar
Pasillo	Activo	Modificar	Eliminar
Sala	Activo	Modificar	Eliminar

© Domotic-UPS 2018-2019

Figura 60. Interfaz estancias.
Elaborado por: Diego Gualpa, José Lagos.

3.4.9. Pantalla de Elementos

En la Figura 61, se visualiza los elementos registrados en el módulo, con el estado correspondiente. Además de un módulo de ingresar estancias y un CRUD que permite modificar o eliminar estancia. Los elementos hacen referencia a los dispositivos conectados dentro de la residencia. De esta forma el usuario se da cuenta en donde se encuentra cada dispositivo y el lugar donde se encuentre activado o desactivado.

Lista de Elementos creados

Lista Elementos			
Nombre Elemento	Lugar Elemento	Tipo Elemento	
Control de Luces	Dormitorio4	Sensor	Modificar Eliminar
Control de luces	Dormitorio3	Actuador	Modificar Eliminar
Control de luces	Dormitorio1	Actuador	Modificar Eliminar
Control de Luces	Dormitorio2	Actuador	Modificar Eliminar
Persianas	Sala	Actuador	Modificar Eliminar
Sensor de humo	Comedor	Sensor	Modificar Eliminar
Sensor de movimiento	Pasillo	Sensor	Modificar Eliminar

© Domotic-UPS 2018-2019

Figura 61. Pantalla elementos
Elaborado por: Diego Gualpa, José Lagos.

3.4.10. Pantalla de Control de Luces

En la Figura 62, se observa las estancias en donde se encuentran ubicados los Modulo Relay para conmutación de luces. Además que por cada lugar donde este asignado el control de luces hay un interruptor que enciende o apaga. Dependiendo del número de lugares ingresados los botones interruptores se irán agregando dinámicamente al CRUD, tomando en cuenta si la estancia creada tiene habitado control de luces.



3.4.11. Pantalla de Sensor de Movimiento

En la Figura 63, se observa la estancia donde se encuentra ubicado el sensor de movimiento y su respectivo estado de detección de presencias dependiendo la variable que el NodeMCU le envíe su estado cambiará a movimiento detectado o no detectado. La notificación que fue enviada desde el NodeMCU cambia el color del resaltado de las notificaciones en la página siendo el color rojo como detectado y verde como no detectado.

Notificación del sensor de movimiento

Lugar	Estado
Pasillo	MOVIMIENTO DETECTADO

© Domotic-UPS 2018-2019

Figura 63. Pantalla de visualización de estado sensor de movimiento.
Elaborado por: Diego Gualpa, José Lagos.

3.4.12. Pantalla de Sensor de Humo

En la Figura 64, se observa la estancia donde se encuentra ubicado el sensor de humo y su respectivo estado.

Notificación del sensor de humo

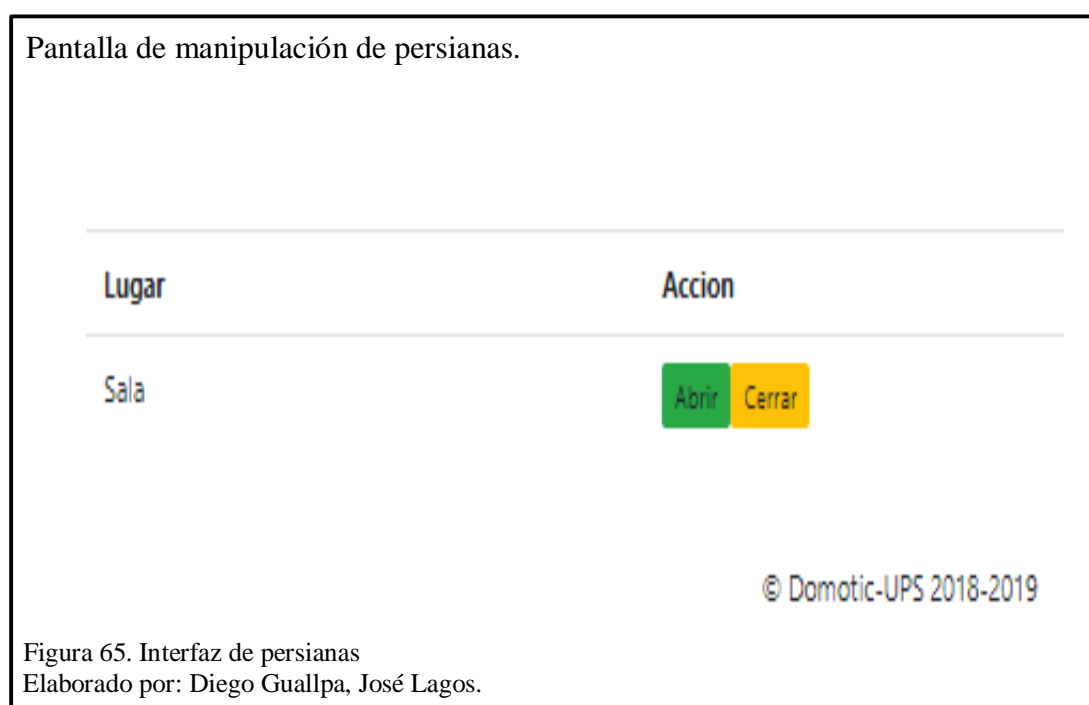
Lugar	Estado
Comedor	NO SE DETECTA PRESENCIA DE HUMO

© Domotic-UPS 2018-2019

Figura 64. Pantalla de visualización de estado de detección de humo.
Elaborado por: Diego Gualpa, José Lagos.

3.4.13. Control de Persianas

En la Figura 65, se observa las estancias en donde se encuentran ubicados los servomotores para controlar persianas. Además que por cada lugar donde este asignado el servomotor hay un botón de subir y bajar persianas.



3.4.14. Pantalla de Auditoria

En la Figura 66, se observa un registro de las acciones realizadas por los diferentes usuarios, estas acciones pueden ser filtradas por fecha, por usuario y por tipo de acción que puede ser ingreso, inicio de sesión, cerrar sesión, ingresar, actualizar y eliminar. La característica principal de esta pantalla es que registra todas las acciones realizadas por los diferentes usuarios ya sean acciones correctas o incorrectas como puede ser el caso de fallo en inicios de sesión, datos no guardados, etc.

Pantalla de Auditoria

Fecha	Usuario	Tipo Acción
<input type="text" value="2019-06-13"/>	<input type="text" value="admin"/>	<input type="text" value="Iniciar sesión"/>

Consultar

Fecha	Usuario	Tipo Acción	Acción
2019-06-13 11:46:20	admin	Iniciar Sesión	Ingresó al sistema
2019-06-13 13:03:14	admin	Iniciar Sesión	Ingresó al sistema
2019-06-13 13:13:56	admin	Iniciar Sesión	Ingresó al sistema
2019-06-13 15:08:32	admin	Iniciar Sesión	Ingresó al sistema
2019-06-13 15:36:03	admin	Iniciar Sesión	Ingresó al sistema
2019-06-13 15:45:32	admin	Iniciar Sesión	Ingresó al sistema

Figura 66. Visualización de auditoria
Elaborado por: Diego Gualpa, José Lagos.

CAPITULO 4

PRUEBAS Y RESULTADOS

Las pruebas realizadas fueron por bloques, en cada uno de los módulos del sistema comprobando su funcionalidad completamente. En cada uno de los módulos se establecieron variables de funcionalidad, así de esta manera se pudo comprobar la eficiencia del sistema al momento de realizar las respectivas pruebas.

En las pruebas realizadas se describen: Pruebas de hardware y pruebas de software.

4.1. Pruebas de Hardware

Control de luces

En la Figura 67, se indica la respuesta que el servidor envía hacia el NodeMCU, se observa el estado de apagado en el foco, las peticiones se originan cuando se da clic en el botón de encendido o apagado. El sistema web envía un uno cuando se enciende o un cero cuando se apaga.

Ejemplo de prueba de toma de datos (iluminación).

```
10
foco 1 apagado
200
10
foco 1 apagado
200
10
foco 1 apagado
200
10
foco 1 apagado
200
10
foco 1 apagado
200
```

Figura 67. Toma de datos del sensor iluminación.
Elaborado por: Diego Gualpa, José Lagos.

En la Figura 68, se muestra el estado de la bombilla cuando se da clic en encender desde el sistema web y el estado de apagado cuando se da clic en apagar. Realizando las pruebas respectivas para los Módulos Relay y encendido de luces.

Relay con bombilla para la prueba



Figura 68. Sensor de luz diseñado.
Elaborado por: Diego Gualpa, José Lagos.

Sensor de movimiento

En la Figura 69, se muestra el valor de estado que determina el sensor ante una presencia. De esta forma envía el estado sea cero o uno hacia la tabla elemento en la cual el sistema web tomará el dato y lo interpretará.

Ejemplo de prueba de toma de datos (movimiento).

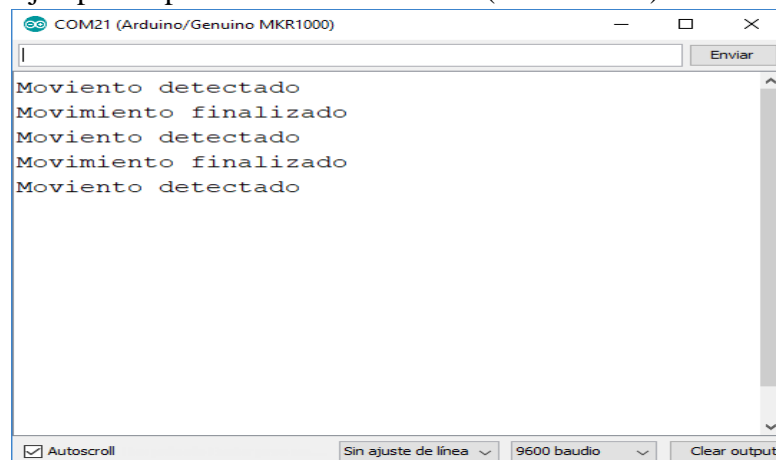
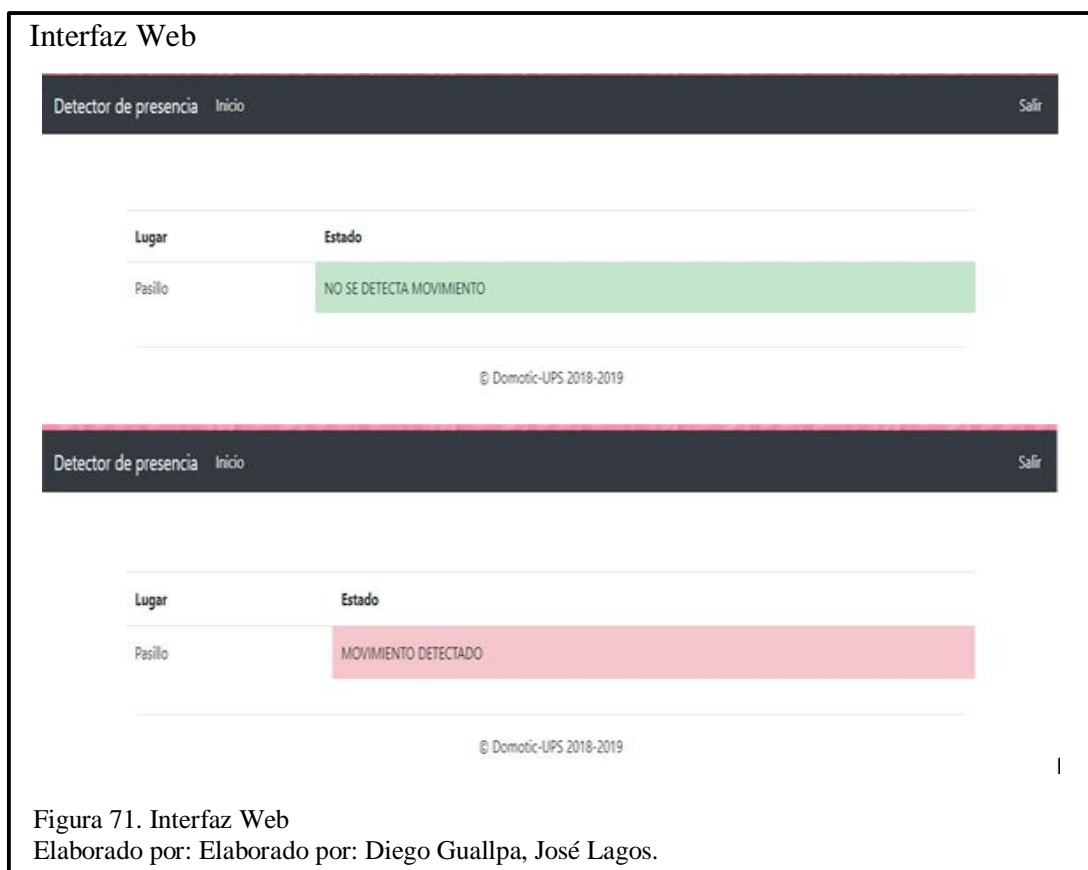


Figura 69. Toma de datos del sensor de movimiento
Elaborado por: Diego Gualpa, José Lagos.

De igual manera se visualizan las pruebas al pasar por el sensor y al dejar de moverse fuera del rango de detección (ver Fig. 70). Así se verifica el funcionamiento del sensor al encenderse o apagarse la bombilla.



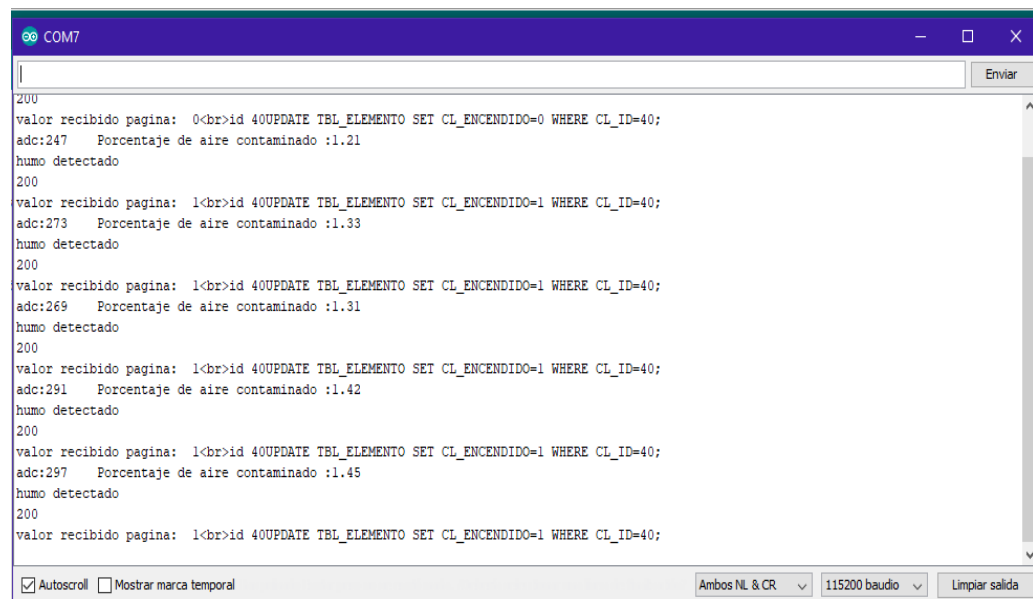
Se puede observar en la Figura 71 que, al momento de enviar el estado del sensor de presencia desde el micro controlador hacia la base de datos. La página realiza la consulta y se muestra el estado del sensor de movimiento, a su vez página se refresca cada 10 segundos y muestra el estado del sensor.



Sensor de humo

En la Figura 72, se muestra los valores de porcentaje de aire contaminado que detecta el sensor de humo, dependiendo el valor el sensor detecta si hay la existencia de humo cuando el valor sobrepasa el uno por ciento de aire contaminado. En la detección de gases en el ambiente el valor de detección puede variar desde cantidades muy pequeñas hasta cantidades muy grandes, depende del tipo de aire que fue detectado. En el caso de la prueba realizada se usó humo conocido científicamente como CO₂, el cuál arroja valores muy altos en su coeficiente de detección. A su vez se verifica como el sensor de humo se encuentra encendido y detectando valores de aire en el ambiente (ver Fig. 73).

Ejemplo de prueba de toma de datos (Humo)



```
200
valor recibido pagina: 0<br>id 40UPDATE TBL_ELEMENTO SET CL_ENCENDIDO=0 WHERE CL_ID=40;
adc:247   Porcentaje de aire contaminado :1.21
humo detectado
200
valor recibido pagina: 1<br>id 40UPDATE TBL_ELEMENTO SET CL_ENCENDIDO=1 WHERE CL_ID=40;
adc:273   Porcentaje de aire contaminado :1.33
humo detectado
200
valor recibido pagina: 1<br>id 40UPDATE TBL_ELEMENTO SET CL_ENCENDIDO=1 WHERE CL_ID=40;
adc:269   Porcentaje de aire contaminado :1.31
humo detectado
200
valor recibido pagina: 1<br>id 40UPDATE TBL_ELEMENTO SET CL_ENCENDIDO=1 WHERE CL_ID=40;
adc:291   Porcentaje de aire contaminado :1.42
humo detectado
200
valor recibido pagina: 1<br>id 40UPDATE TBL_ELEMENTO SET CL_ENCENDIDO=1 WHERE CL_ID=40;
adc:297   Porcentaje de aire contaminado :1.45
humo detectado
200
valor recibido pagina: 1<br>id 40UPDATE TBL_ELEMENTO SET CL_ENCENDIDO=1 WHERE CL_ID=40;
```

Figura 72. Toma de datos del sensor de movimiento
Elaborado por: Diego Gualpa, José Lagos.

Sensor de Humo MQ-135

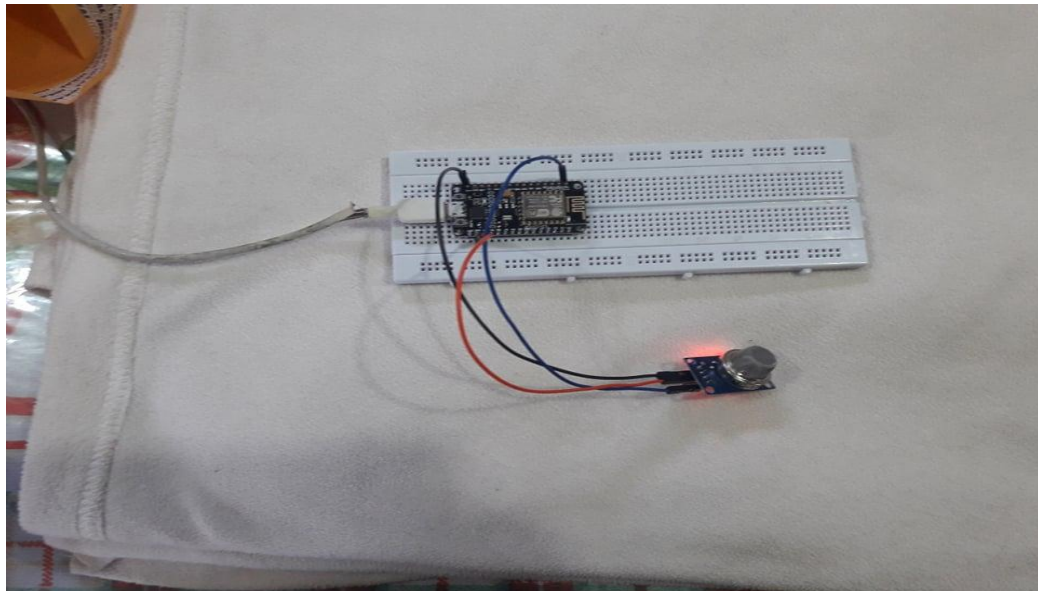


Figura 73. Sensor de Humo conectado al NodeMCU
Elaborado por: Diego Gualpa, José Lagos

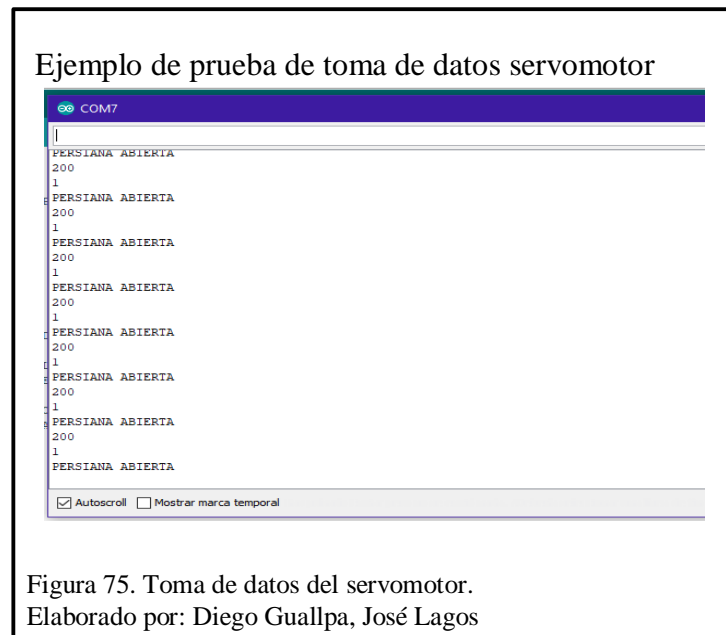
El NodeMCU envía por método HTTP GET el valor del estado del sensor y actualiza los campos correspondientes en la tabla elemento para obtener el estado en la página del sistema.

En la Figura 74, se visualiza el estado del sensor dependiendo el valor que envíe el NodeMCU. Esta página se refresca cada diez segundos enviando consultas hacia la base de datos para obtener el estado.

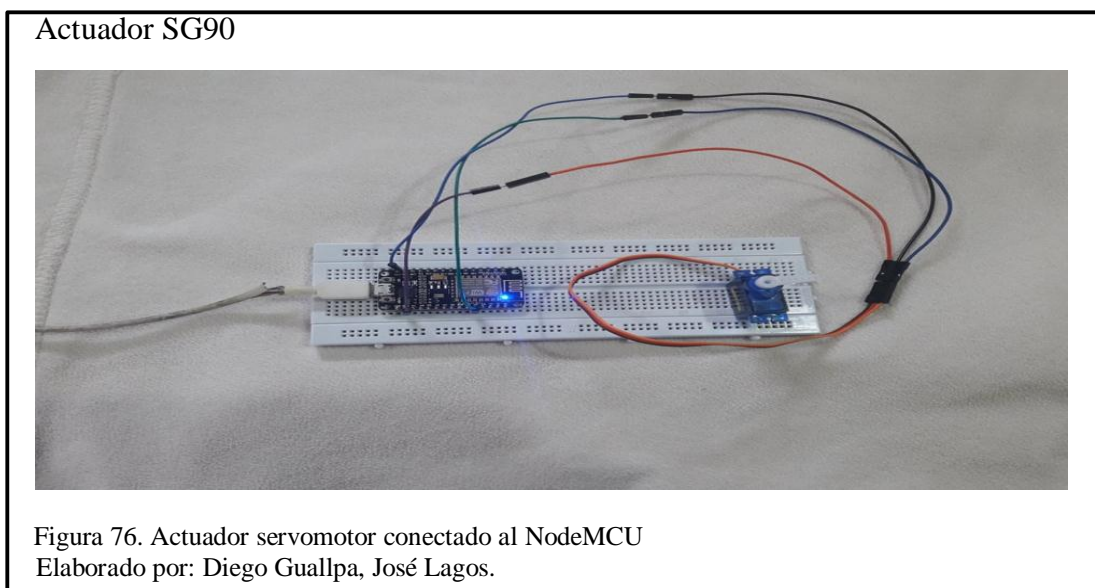


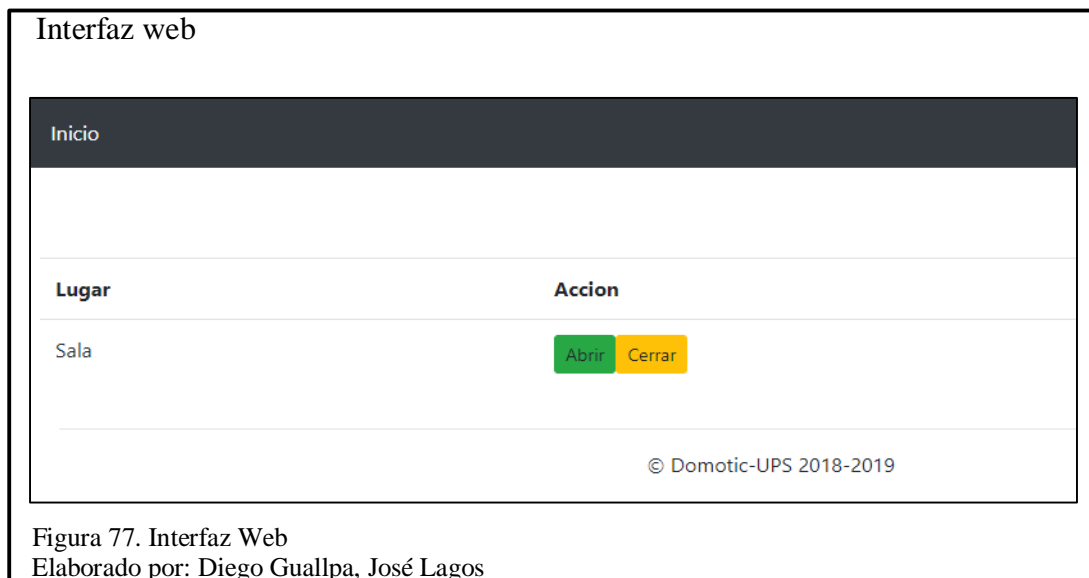
Servomotor

En la figura 75, se muestra la acción que se envía desde la página web hacia el servomotor en este caso cuando el NodeMCU realiza la petición al servidor obtiene el uno que activa el motor y lo hace girar a 180 grados y en caso contrario cuando obtiene el cero el motor gira de retorno a cero grados.



En la Figura 76, se muestra que el motor gira en sentido hacia la derecha cuando se activa el boton de abrir,





En la Figura 77, se indica los controles del servomotor desde la página web con los botones de abrir y cerrar.

4.2. Pruebas de software

Las diferentes pruebas de software se realizaron pruebas en las ventanas más importantes de sistema, así se pudo comprobar los funcionamientos básicos de cada una de ellas.

Acceso a usuarios



En la pantalla de acceso de usuarios se muestra el acceso exitoso hacia la página de inicio en donde se realizará el control de los dispositivos interconectados. Las acciones que se realizan dentro del sistema quedan registradas en la auditoria del sistema web, el evento que acaba de registrarse es el ingreso a la plataforma.

En la figura 79, se realizan las pruebas de validación de campos en la ventana de ingreso de usuarios. Se validan que los campos no estén vacíos y el correo electrónico debe ser correcto en su formato.

Ingreso de Usuarios

Ingreso de usuarios

Alias Usuario

Ingresar el nombre del usuario

Alias de usuario

Nombre Usuario

Ingresar el nombre del usuario

Nombre y apellido de usuario

Correo Electrónico

Ingresar el correo electrónico

Correo Electrónico

Tipo

Escoger un perfil

---Elija una opción---

Figura 79. Validación de campos vacíos en usuario.
Elaborado por: Diego Gualpa, José Lagos.

En la Figura 80, se observa el mensaje de confirmación cuando el registro ha sido ingresado con éxito. Se debe recalcar que este estado de ingreso correcto se aplica para modificar datos actualizar o eliminar.

Usuario exitoso

Usuario ingresado exitosamente.

Alias Usuario

Alias de usuario

Nombre Usuario

Nombre y apellido de usuario

Correo Electrónico

Correo Electrónico

Tipo

----Elija una opción----

Contraseña

Password

Guardar

Borrar campos

Figura 80. Usuarios ingresados
Elaborado por: Diego Gualpa, José Lagos

Se verifica que el usuario ingresado anteriormente esta correcto y se puede visualizar en la consulta general de los usuarios ingresados (ver Fig. 81).

Listado de usuarios ingresados

Lista Usuarios			
Nombre	Perfil	Correo	Opciones
Administrador	Administrador	josen890_la@hotmail.com	<div>ModificarEliminar</div>
Diego Gualpa	Administrador	dier1989@hotmail.com	<div>ModificarEliminar</div>
José Lagos	Controlador	josen890_la@hotmail.com	<div>ModificarEliminar</div>
RENATO MINA	Controlador	renatom@gmail.com	<div>ModificarEliminar</div>

© Domotic-UPS 2018-2019

Figura 81. Consulta de usuarios.
Elaborado por: Diego Gualpa, José Lagos

De igual manera como todos los módulos se han grabado las acciones realizadas en auditoria haya registrado la acción de insertar en la tabla relacionada con auditoria. Al momento de modificar el usuario se debe ubicar en la fila que se desea modificar los campos se da clic en modificar y automáticamente se carga la información en los campos de insertar.

En la Figura 82, se visualiza el CRUD luego de haber modificado un usuario existente.

Usuarios que seran modificados.

Lista Usuarios			
Nombre	Perfil	Correo	Opciones
Administrador	Administrador	josen890_la@hotmail.com	Modificar Eliminar
Diego Guallpa	Administrador	dier1989@hotmail.com	Modificar Eliminar
José Lagos	Controlador	josen890_la@hotmail.com	Modificar Eliminar
RENATO MINA	Controlador	renatojulio@hotmail.com	Modificar Eliminar
© Domotic-UPS 2018-2019			

Figura 82. Listas de usuarios existentes.
Elaborado por: Diego Guallpa, José Lagos.

En la Figura 83, Se ubica en los campos a modificar y a continuación se debe dar clic en guardar. Después de insertar, modificar o eliminar los registros al instante se visualizara el CRUD actualizándose en cada una de sus filas.

Modificar usuario

Alias Usuario

Alias de usuario

Nombre Usuario

Nombre y apellido de usuario

Correo Electrónico

Correo Electrónico

Tipo

----Elija una opción----

Contraseña

Password

Guardar

Borrar campos

Figura 83. Modificación de usuarios
Elaborado por: Diego Gualpa, José Lagos.

Después de realizar acciones en el sistema se realiza la verificación correspondiente en la auditoria. El usuario que ha sido modificado y el campo que modificó así hacen que la auditoria se registre con más efectividad.

En la Figura 84, se verifica la acción de modificar usuario dentro de la auditoria con la fecha de actualización del registro.

Consulta de usuario modificado.

Fecha Usuario Tipo Acción

2019-06-12 Todos Modificar

Consultar

Fecha	Usuario	Tipo Acción	Acción
2019-06-12 11:16:23	admin	Modificar	Modificó usuario RENATO MINA campo correo
2019-06-12 11:27:22	admin	Modificar	Modificó estancia
2019-06-12 11:32:19	admin	Modificar	Modificó elemento
2019-06-12 11:34:47	admin	Modificar	Modificó elemento

© Domotic-UPS 2018-2019

Figura 84. Verificar usuario modificado
Elaborado por: Diego Gualpa, José Lagos.

En la figura 85, se muestra la manera de como eliminar los usuarios en la tabla, se selecciona la fila y se da clic en eliminar. De la misma forma como las ventanas anteriores de refresca el listado del CRUD después de eliminar.

Se toma en cuenta que al eliminar un registro lo hace por medio de índices implementados dentro del campo id del registro. Cada índice es granular por lo que la tarea de eliminar registros es rápida y segura.

Usuario eliminado.

Usuario eliminado correctamente.

Controlador ▼

Contraseña

.....

Guardar **Borrar campos**

Lista Usuarios

Nombre	Perfil	Correo	Opciones
Administrador	Administrador	josen890_la@hotmail.com	Modificar Eliminar
Diego Guallpa	Administrador	di.er1989@hotmail.com	Modificar Eliminar
José Lagos	Controlador	josen890_la@hotmail.com	Modificar Eliminar

© Domotic-UPS 2018-2019

Figura 85. Eliminación de usuarios.
Elaborado por: Diego Guallpa, José Lagos.

De la misma manera la acción de eliminar usuarios es registrada en la auditoria. Se registra con el nombre del usuario que fue eliminado (ver Fig. 86).

Auditoria de usuarios eliminados.

Fecha Usuario Tipo Acción

2019-06-12 Todos ▼ Eliminar ▼

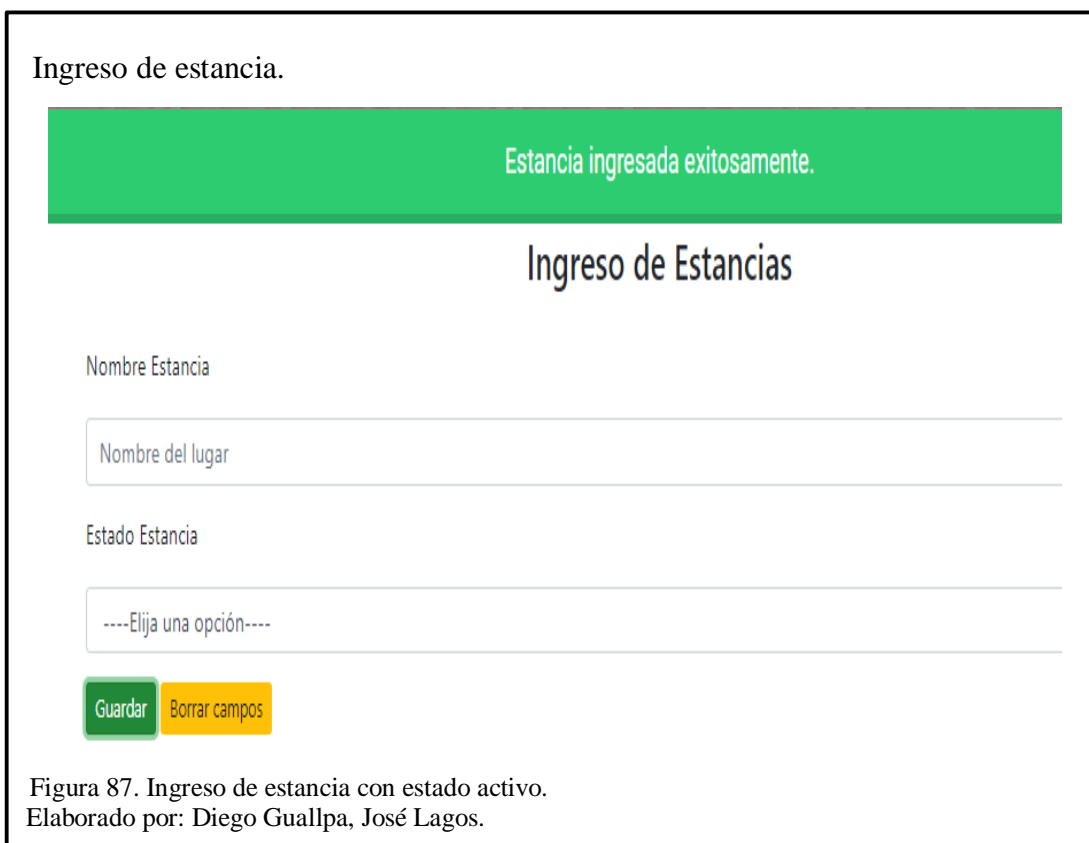
Consultar

Fecha	Usuario	Tipo Acción	Acción
2019-06-12 11:16:48	admin	Eliminar	Dió de baja usuario RENATO MINA

Figura 86. Auditoría usuarios eliminados.
Elaborado por: Diego Guallpa, José Lagos.

Estancias

En la figura 87, se visualiza la parte de ingresar estancias, se ingresa los campos correspondientes y se debe dar clic en guardar, el mensaje de éxito se visualiza automáticamente.



The screenshot displays a web form titled "Ingreso de Estancias". At the top, a green banner contains the text "Estancia ingresada exitosamente." Below this, the form has two input fields: "Nombre Estancia" with a placeholder "Nombre del lugar" and "Estado Estancia" with a placeholder "----Elija una opción----". At the bottom of the form are two buttons: a green "Guardar" button and a yellow "Borrar campos" button. The entire form is enclosed in a black border.

Ingreso de estancia.

Estancia ingresada exitosamente.

Ingreso de Estancias

Nombre Estancia

Nombre del lugar

Estado Estancia

----Elija una opción----

Guardar Borrar campos

Figura 87. Ingreso de estancia con estado activo.
Elaborado por: Diego Guallpa, José Lagos.

Elementos

Se ingresan los elementos que irán en cada estancia registrada, estos pueden ser sensores o actuadores. Se debe escribir en cada uno de los campos correspondientes y se debe dar clic en guardar. El mensaje de confirmación exitosa aparecerá (ver Fig. 88).

Elemento ingresado

Elemento ingresado exitosamente.

Ingreso de Elementos

Lugar

----Elija una opción----

Nombre Elemento

Nombre del Objeto

Tipo Elemento

----Elija una opción----

Guardar

Borrar campos

Figura 88. Ingreso de elemento.
Elaborado por: Diego Gualpa, José Lagos.

4.3. Presupuesto

A continuación, se detalla un resumen estimado de todos los recursos utilizados en el desarrollo del proyecto en la Tabla 6. Con el presupuesto calculado se verifica según estudios realizados que el costo es relativamente bajo en comparación a otros proyectos de domótica antes realizados.

Tabla 6. Presupuesto.

Descripción	Cantidad	Precio Unitario	Precio Total
NodeMCU ESP8266	3	8,00	24,00
Kit Raspberry PI 3	1	112,00	100,00
Accesorios		20,00	20,00
Sensor PIR	1	5,00	5,00
Sensor Humo	1	5,00	5,00
Relay 4ch	1	6,00	6,00
Relay 1ch	1	2,00	2,00
Servo Motor	2	5,00	5,00
Software aplicado		00,00	00,00
Costo Final			\$178 USD.

Fuente: Diego Gualpa, José Lagos

Nota: En la presente tabla se puede visualizar el costo estimado de cada uno de los dispositivos utilizados en el proyecto, tanto a nivel de hardware y software.

Con el proyecto realizado se mejora la calidad de vida de los usuarios con la automatización de tareas básicas del hogar. A su vez puede ayudar a personas que tengan problemas de discapacidad ya que desde un celular o Tablet se puede controlar los dispositivos mediante el sistema realizado.

CONCLUSIONES

- Los sensores y actuadores utilizados permitieron la fácil interacción con el servidor doméstico, por medio del protocolo http y direccionamiento IPV4 la comunicación logró un grado de efectividad del 98%.
- Se diseñó, configuró e implementó el prototipo mediante una red de sensores ubicados en diferentes partes de la maqueta a escala obteniendo una efectividad del 97,2%, de acuerdo a los resultados obtenidos.
- Una vez realizadas las pruebas basándose en el diagrama electrónico inicial del prototipo, se produjeron daños en varios componentes electrónicos por lo que se hizo un rediseño del diagrama electrónico realizando las conexiones adecuadas y distribuciones de voltaje correspondientes.

RECOMENDACIONES

- Con respecto a la conexión de los circuitos electrónicos se recomienda que se conozca previamente los voltajes de entrada y salida de cada dispositivo para evitar que el dispositivo tenga una mal funcionalidad.
- Se recomienda que para investigaciones futuras se explote las potencialidades de la tarjeta de desarrollo MCU ESP8266, que posee versatilidad y bajo costo.
- Ampliar las funcionalidades del prototipo con mayor cantidad de sensores y actuadores.
- Se recomienda utilizar el sistema operativo Raspbian stretch en modo de consola debido que esto hace que la efectividad en los procesos de kernel sea menos costosa en recursos de hardware.

LISTA DE REFERENCIAS

- Kawashima. (1 de Enero de 2019). *Blog de Tecnología Kokensha*. Obtenido de <https://kokensha.xyz/raspberry-pi/what-can-raspberry-pi-do-2019/>
- Mejia Jervis, T. (2017). *Lifede*. Recuperado el 17 de Julio de 2019, de <https://www.lifeder.com/tipos-de-sensores/>
- AbcElectrónica. (2015). *ABC Proyectos Electrónicos*. Recuperado el 22 de 05 de 2019, de ABC Proyectos Electrónicos: <https://www.abcelectronica.net/productos/relas/modulos/>
- Abellán, M. (2019). *Programa Ergo Sum*. Recuperado el 12 de Mayo de 2019, de <https://www.programoergosum.com/cursos-online/raspberry-pi/232-curso-de-introduccion-a-raspberry-pi/instalar-raspbian>
- AliExpress.com*. (Abril de 2019). Recuperado el 29 de Abril de 2017, de https://www.aliexpress.com/item/10pcs-lot-SG90-9g-Mini-Micro-Servo-for-RC-Planes-Fixed-wing-Aircraft-model-telecontrol-250/33028428583.html?aff_platform=aaf&cpt=1560578276099&onelink_item_from=33028428583&onelink_thrd=0.015&onelink_page_from=ITEM_DETAIL&o
- Arciniega, J. L. (2017). *Benchmark de selección de sensores para una WSN de recolección de datos para un sistema de alerta temprana de incendios forestales*. Universidad Técnica del Norte, Ingeniería en Electrónica y Redes de Comunicación, Ibarra.
- Arévalo, P. L. (2017). *Desarrollode plataforma de Internet de las cosas para la toma de decisiones*. Universidad Distrital Francisco José de Caldas, Facultad Tecnológica, Bogotá.
- Arqcompus-domotica. (27 de Junio de 2009). *Domotica*. Recuperado el 23 de Mayo de 2019, de <http://arqcompus-domotica.blogspot.com/2009/06/arquitectura.html>

- Banggood. (06 de 05 de 2016). *Banggood*. Recuperado el 26 de 05 de 2019, de Banggood: <https://es.banggood.com/3-In-1-Raspberry-Pi-3-Model-B-Official-Case-Heatsinks-Set-reviews-p1048292.html>
- Barela, M. (4 de Julio de 2018). *Explore & Learn*. Recuperado el 19 de Julio de 2019, de <https://learn.adafruit.com/make-it-move-with-crickit/standard-servo-motors>
- Bembibre, V. (Enero de 2009). *Definición ABC*. Recuperado el 7 de Mayo de 2019, de Definición ABC: <https://www.definicionabc.com/tecnologia/html.php>
- Bench, H. (2015). *Arduino HC-SR501 Motion Sensor Tutorial*. Recuperado el 18 de Julio de 2019, de <http://henrysbench.cpnfatz.com/henrys-bench/arduino-sensors-and-input/arduino-hc-sr501-motion-sensor-tutorial/>
- Blog de Domotica*. (29 de Abril de 2015). Recuperado el 12 de Abril de 2019, de <http://serconint.com/blog/sensores-para-domotica/>
- Cabeza, P. J. (2018). *Vivienda inteligente: proyecto domótico de bajo coste para una vivienda unifamiliar*. Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación, Universidad de Cantabria, España.
- Cahuna, & Pari, R. E. (2017). *Sistema de control de tráfico vehicular en base A*. Universidad Mayor de San Andres, Facultad de Ciencias Puras y Naturales Carrera de Informática, Bolivia.
- Ceja José, R. R. (2017). *Módulo ESP8266 y sus aplicaciones en el internet de las cosas*. Lima: Revista de Ingeniería Eléctrica.
- Chicaiza, R. (2016). *Diseño e implementación de un sistema eléctrico inteligente, en una oficina modelo basada en la aplicación de la domótica*. Universidad Técnica de Cotopaxi, Unidad Académica de Ciencias de la Ingeniería y Aplicadas, Latacunga.
- Cuzco Carrion, G. L., & Layana Vallejo, W. A. (2012). *Sistema de alarma automotizacion y control de equipos a distancia a través de línea telefonica y pagina WEB*. Guayaquil: (Carrera de Ingeniería Electrónica con mención de sistemas industriales).Universidad Politécnica Salesiana.

- DiYMakers. (21 de Julio de 2014). *DiYMakers*. Recuperado el 28 de Abril de 2019, de DiYMakers: <http://diymakers.es/usando-el-puerto-gpio/>
- Domótica Vento. (25 de 02 de 2017). *Partes de un sistema Domótico*. Obtenido de <http://sistemasdomoticos.com/partes-de-un-sistema-domotico/>
- Electrónica, M. (2016). *MK Electrónica*. Recuperado el 28 de Mayo de 2019, de MK Electrónica: <https://mkelectronica.com/producto/microlab-plataforma-educativa-tarjetas-controladoras/>
- Eufemio, P. R. (2008). *Diseño y Simulación de un Sistema Domótico para una Residencia*. Escuela Técnica Superior de Ingeniería de Telecomunicación, Ingeniería de Telecomunicación, Colombia.
- Ganchoso, J. M. (2018). *Prototipo de sistema de monitoreo de CO2 en exteriores por medio de hardware*. Universidad de Guayaquil, Académico de Graduación. Guayaquil: Facultad de Ingenieria Industrial.
- Garcia, A. T. (2016). *Diseño e implementación de una estación*. Maestria en Telecomunicaciones, Universidad Abierta de Cataluña, Electrónica, España.
- GitHub,Inc . (01 de 10 de 2016). Recuperado el 2019 de 05 de 15, de GitHub,Inc : <https://github.com/jaimelaborda/Planta-Twittera/wiki/1.-Introducci%C3%B3n-al-ESP8266-y-NodeMCU>
- González, A. G. (02 de 12 de 2016). *PanamaHitek*. Recuperado el 16 de 05 de 2019, de <http://panamahitek.com/que-es-y-como-funciona-un-servomotor/>
- Harrington, W. (2015). Raspbian. En W. Harrington, *Aprendiendo Raspbian* (págs. 10-15). Packt Publishing Ltd.
- Hernández Pérez, I. N. (2016). *Desarrollo de un sistema de control Domótico utilizando un microcontrolador y tecnologías PLC*. Tesis , Instituto Politécnico Nacional , Ingeniería Eléctrica, Mexico.
- Hernández, L. d. (2016). *ProgramarFacil.com*. Recuperado el 03 de 05 de 2019, de ProgramarFacil.com: <https://programarfacil.com/tutoriales/fragmentos/servomotor-con-arduino/>

- Inés, J. R. (2017). *Sistema domótico hardware y software basado en un sistema embebido y una aplicación móvil*. Madrid: Dpto. Ingeniería Informática.
- J.Rold, P. R. (25 de Agosto de 2017). *QUORA*. Recuperado el 28 de Mayo de 2019, de <https://es.quora.com/Qu%C3%A9-son-los-actuadores-y-qu%C3%A9-tipos-hay-en-el-internet-de-las-cosas>
- Jáimez, S. (2 de Enero de 2017). *Todo para el Automatismo y Electrónica*. Recuperado el 27 de Mayo de 2019, de <https://akronelectronica.com/blog/ingenieria-en-el-hogar/los-beneficios-la-domotica-hogar/>
- Jorge, R. A. (2017). *Implementacion de un sistemas de administración energética mediante raspberry pi 3, bajo las condiciones de la norma ISO-5001 aplicada a cargas domésticas*. Proyecto de Titulación, Universidad Politécnica Salesiana, Carrera de Ingeniería Electrica, Guayaquil.
- José Félix, A. G. (2012). *Plataforma domótica basada en la integración de un hipervisor*. Mexico: Universidad de Colima.
- Koyanagi, F. (5 de Febrero de 2018). *Instructables.com*. Recuperado el 5 de Mayo de 2019, de <https://www.instructables.com/id/NodeMCU-ESP8266-Details-and-Pinout/>
- Laboratorio de herramientas*. (11 de Noviembre de 2015). Recuperado el 22 de Julio de 2019, de <https://tool-lab.com/make/raspberrypi-startup-22/>
- Llamas, L. (21 de Octubre de 2016). *Ingeniería, informática y diseño*. Recuperado el 27 de Mayo de 2019, de Ingeniería, informática y diseño: <https://www.luisllamas.es/arduino-detector-gas-mq/>
- LLamas, L. (1 de Junio de 2018). *Ingeniería, informática y diseño*. Recuperado el 28 de Mayo de 2019, de <https://www.luisllamas.es/esp8266-nodemcu/>
- Malášek, J. (26 de Julio de 2011). *Pololu Robotics and Electronics*. Recuperado el 20 de Julio de 2019, de <https://www.pololu.com/blog/24/continuous-rotation-servos-and-multi-turn-servos>
- Marian, P. (2 de Febrero de 2015). *Electro Schematics*. Recuperado el 22 de Julio de 2019, de <https://www.electroschematics.com/11276/esp8266-datasheet/>

- Martinez, V. (2015). *Sistema de Domótica para control y supervisión de una habitación de manera remota*. Pontificia Universidad Javeriana, Ingeniería Electrónica, Bogotá.
- Mecafenix, F. (11 de Abril de 2017). *Ingeniería Mecafenix*. Recuperado el 12 de Mayo de 2019, de <https://www.ingmecafenix.com/automatizacion/sensores/>
- Mecaxis.Cat. (2007). Recuperado el 19 de Julio de 2019, de Sensores y Actuadores: http://mecaxis.cat/wp-content/uploads/2009/01/sensores_y_actuadores_iav_tema2.pdf
- Mendoza, P. (25 de Abril de 2018). *Xataka*. Recuperado el 30 de Abril de 2019, de Xataka: <https://www.xataka.com/ordenadores/raspberry-pi-3-model-b-analisis-mas-potencia-y-mejor-wifi-para-un-minipc-que-sigue-asombrando>
- Mera, Y. (14 de 03 de 2018). *Xataka*. Obtenido de Xataka: <https://www.xataka.com/ordenadores/llega-la-nueva-raspberry-pi-3-model-b-mismo-precio-pero-mas-velocidad-y-wifi-de-doble-banda>
- Mono Wireless*. (2015). Recuperado el 15 de Julio de 2019, de <https://mono-wireless.com/jp/tech/wsn.html>
- Nava Oscar, R. E. (2016). *Servidor Http seguro en un sistema embebido*. Mexico: Universidad Autónoma Metropolitana, Unidad Azcapotzalco.
- NaylampMechatronics*. (2018). Recuperado el 15 de Mayo de 2019, de <https://naylampmechatronics.com/sensores-gas/73-sensor-calidad-aire-mq135.html>
- Ordoñez Garcia, J. E. (2017). *Diseño de Interfaz usando minicomputador Raspberry Pi a ser usada en la impresion en 3 Dimensiones*. Tesis de Grado, Escuela Superior Politécnica del Litoral, Facultad de Ingeniería en Electricidad y Computación , Guayaquil.
- Parque de Innovación Tecnológica. (2015). *Parque de Innovación Tecnológica*. Recuperado el 12 de 05 de 2019, de <http://innovacion.uas.edu.mx/5062/>
- Politécnica, B. (2012). *Memorias XII Semana de la Ciencia*. Madrid: Biblioteca Politécnica.

- Porto, J. P., & Gardey, A. (2013). *Definición de protocolo de comunicación*.
Obtenido de <https://definicion.de/protocolo-de-comunicacion/>
- Redes de Sensores Inalámbricos WSN*. (16 de Julio de 2012). Recuperado el 23 de Mayo de 2019, de <https://informaticaredes2012.blogspot.com/>
- Rodrigo, J. (25 de Octubre de 2016). *Panamahitek*. Recuperado el 29 de Mayo de 2019, de <http://panamahitek.com/esp8266-y-nodemcu-la-nueva-generacion/>
- Rodríguez, J. A. (2017). *Sistemas en Tiempo Real para Sistemas Embebidos*. Cuba: Universidad Central “Marta Abreu” de Las Villas.
- Rodriguez, L. (23 de Junio de 2014). *La Domótica*. Recuperado el 22 de Mayo de 2019, de <http://laudomotica.blogspot.com/2014/06/de-donde-proviene-la-palabra-domotica.html>
- Rodríguez, R. J., & Quito, P. (2017). *Prototipo de telecontrol de una red inalámbrica de sensores para seguridad y acciones básicas del hogar, aplicado a personas con discapacidad*. Escuela Superior Politécnica del Chimborazo, Ingeniería en Electrónica, Telecomunicaciones y Redes, Riobamba.
- Rojas Bravo, M. A., & Tua Guerrero, S. M. (2018). *Desarrollo de un sistema de monitorización de amenazas físicas con sistemas embebidos y red de sensores (WSN), para el data center del laboratorio de servidores de la carrera de computación de la universidad politécnica salesiana campus sur*. Universidad Politécnica Salesiana, Ingeniería de Sistemas, Quito.
- Rubén, V. (24 de Febrero de 2019). *Softzone*. Recuperado el 22 de Abril de 2019, de <https://www.softzone.es/2019/02/24/mejores-sistemas-raspberry-pi-3-2019/>
- Ruiz Gutiérrez, J. M. (2007). *Manual Programacion Arduino*. Obtenido de <https://arduinoobot.pbworks.com/f/Manual+Programacion+Arduino.pdf>
- Salas Arriarán, S. (2015). *Todo sobre sistemas embebidos*. Lima: Universidad Peruana de Ciencias Aplicadas.
- Santos, R. (15 de Enero de 2019). *Random Nerd Tutorials*. Recuperado el 20 de Julio de 2019, de <https://randomnerdtutorials.com/guide-for-relay-module-with-arduino/>

- Sejuku*. (16 de Junio de 2016). Recuperado el 22 de Julio de 2019, de
<https://www.sejuku.net/blog/6026#HTML>
- Sensorica*. (28 de Octubre de 2016). Recuperado el 15 de Mayo de 2019, de
Sensorica: <https://sensoricanumpaue.blogspot.com/2016/10/tipos-de-sensores.html>
- Sistemas.com*. (s.f.). Obtenido de <https://sistemas.com/html.php>
- Smelpro. (01 de 11 de 2016). *Smelpro*. Recuperado el 13 de 05 de 2019, de Smelpro:
https://www.smelpro.com/tienda/wtblog/5_Arduino-Servomotor-SG90.html
- The Valley*. (11 de Abril de 2016). Recuperado el 22 de Abril de 2019, de
<https://thevalley.es/blog/sensores-los-dispositivos-alimentan-iot/>
- Torres Calvo, F. J. (2014). *Analisis y Diseño de una red domótica*. Recuperado el 12 de Mayo de 2019, de
<http://cybertesis.uach.cl/tesis/uach/2014/bmfcic169a/doc/bmfcic169a.pdf>
- Universidad Mariano Gomez. (Mayo de 2016). *Introducción a Sistemas de Computo*.
Recuperado el 23 de Mayo de 2019, de
<https://itzelramirezblog.wordpress.com/domotica/>
- València, U. P. (18 de 12 de 2013). *Museo de Informática*. Obtenido de Museo de
Informática: <http://histinf.blogs.upv.es/2013/12/18/raspberry-pi/>
- Workship Magazine*. (3 de Julio de 2017). Obtenido de
https://goworkshop.com/magazine/what_can_raspberrypi_do/